

[< Go to the original](#)

### Continue with all changes?

You made 16 changes to the company details. Continue to save all changes. Discard to remove all changes.

[Cancel](#)[Discard](#)[Save and Continue](#)

# Why "Cancel" should be a link, and not a button

Part 2: your feedback. About SPAs, hyperlinks, dark patterns, and accessibility.

**Karim Maassen**[Follow](#)

 UX Collective a11y-light ~35 min read · June 28, 2023 (Updated: June 30, 2023) · Free: No

My article ([Should "cancel" be a button or a link?](#)) stated that when users are presented with a "cancel" option, this should be a link, and not a button. It received feedback which I talk about in this follow-up.

### Should “cancel” be a button or a link?

Which one is UX best practice?

uxdesign.cc

# Why even discuss this subject?

fine". And sure enough, having a *button* instead of a *link* where we believe a link improves your usability, will probably not leave your users at a *complete* loss to ruin their day.

However, by being consistent and predictable in your patterns, users *will* be empowered while interacting with your solution. This is a good thing as it adds to your user's confidence as they traverse your solution.

Person on the internet

**Why does this even matter?**

Me

**The discussion regarding *buttons vs. links* is not about styling, it's about usability. It's about reinforcing patterns.**

Someone asked: "Why does this even matter?" Because it's about usability!

## Improve when we can!

It also doesn't take away from the fact we should always strive to improve usability when we see an opportunity. At the very least, discuss it. Especially when we believe there is sound reasoning behind it. The discussion regarding *buttons vs links* is not about styling, it's about usability; it's about reinforcing patterns.

**uxdworld:** Great user experience (UX) design allows users to understand it and interact with it instantly. It is designed in a way that users can predict their next interaction and its outcome or response.

**ab11y:** Patterns enable people to learn, relate and engage as their comprehension is built upon all previous interactions with particular

Dr. Ralf Speth wrote: If You Think Good Design is Expensive, You Should Look At The Cost of Bad Design

H Locke wrote: Design patterns could change over time in terms of user familiarity rather than being successfully designed in line with cognitive processes. This is the case with the hamburger menu for example

## The Feedback

A large part of the feedback was constructive and insightful. Which really goes to show how important and fruitful it can be putting your ideas out there for the world to see, and why a community like this exists. This follow-up article addresses the feedback the previous article received. I will try and respond to the most common types of responses, and eventually try to draw some conclusions.

Now let's dive into it. We can categorize the feedback into several somewhat distinct types. I've lumped them together as follows:

### "Button, no discussion"

Person on the internet

**Button. What else?!**

Me

**Well...**

Some of the feedback was just "*button, what else?!*" — Of course, the title of the original article had us set up for a single-worded answer, so we're partly in the

Maybe these reactions came from those who are convinced that "*button*" simply is the correct pattern, and feel reading the article would not convince them otherwise. Or maybe they think that "*button*" in fact is the correct pattern, *despite* having read the article and were simply not convinced. For some, it's just a non-question, regardless of the existence of other visions. Or, as [Simon Spichak](#) so beautifully says: "*humans are irrational*".

There is a catch though...

### **Obvious isn't obvious**

Most UI we design comes across as "obvious" to the user. Or at least, that's our intention. However, most of those designs went through a lot of (unseen to the user) thought process and research. So it's **obvious to most, but a complex problem for the designer**. And that's where it's getting interesting. It's where the — for some — heated controversy, or the "*no discussion*" stance comes from. While it *seems* obvious, in reality it actually might not be.

A discussion I had in the past about table rows and their interaction, is a good example of this phenomenon. Consider the following table. A common question that is often raised when looking at an image like this, is "*how does a user navigate to the detail page of a company in this table?*"

Company	Location	Employees	CEO
Apple	<a href="#">Cupertino, CA</a>	147,000	<a href="#">Tim Cook</a>
Microsoft	<a href="#">Redmond, WA</a>	166,475	<a href="#">Satya Nadella</a>
Alphabet	<a href="#">Mountain View, CA</a>	135,301	<a href="#">Sundar Pichai</a>
Amazon	<a href="#">Seattle, WA</a>	575,700	<a href="#">Jeff Bezos</a>
Disney	<a href="#">Burbank, CA</a>	201,000	<a href="#">Bob Chapek</a>

"how does a user navigate to the detail page of a company?"

Seeing this table design, a typical stakeholder would go into discussion with the UI designer, asking that exact same question: *"how does a user navigate to the detail page?"* — Trying to solve that self-proclaimed usability issue, they would then suggest a button or text which says *"click here to go to the detail view"* (or something similar). The motivation being *"how would a user know without it?"* Frustratingly, the UI designer would then come up with UI like this:

Company	Location	Employees	CEO	
Apple	<a href="#">Cupertino, CA</a>	147,000	<a href="#">Tim Cook</a>	<a href="#">Click to see details →</a>
Microsoft	<a href="#">Redmond, WA</a>	166,475	<a href="#">Satya Nadella</a>	<a href="#">Click to see details →</a>
Alphabet	<a href="#">Mountain View, CA</a>	135,301	<a href="#">Sundar Pichai</a>	<a href="#">Click to see details →</a>
Amazon	<a href="#">Seattle, WA</a>	575,700	<a href="#">Jeff Bezos</a>	<a href="#">Click to see details →</a>
Disney	<a href="#">Burbank, CA</a>	201,000	<a href="#">Bob Chapek</a>	<a href="#">Click to see details →</a>

Company	Location	Employees	CEO	
Apple	<a href="#">Cupertino, CA</a>	147,000	<a href="#">Tim Cook</a>	<a href="#">Details →</a>
Microsoft	<a href="#">Redmond, WA</a>	166,475	<a href="#">Satya Nadella</a>	<a href="#">Details →</a>
Alphabet	<a href="#">Mountain View, CA</a>	135,301	<a href="#">Sundar Pichai</a>	<a href="#">Details →</a>
Amazon	<a href="#">Seattle, WA</a>	575,700	<a href="#">Jeff Bezos</a>	<a href="#">Details →</a>
Disney	<a href="#">Burbank, CA</a>	201,000	<a href="#">Bob Chapek</a>	<a href="#">Details →</a>

Needless and obnoxious "click here" links or buttons

Company	Location	Employees	CEO
<a href="#">Apple</a>	<a href="#">Cupertino, CA</a>	147,000	<a href="#">Tim Cook</a>
<a href="#">Microsoft</a>	<a href="#">Redmond, WA</a>	166,475	<a href="#">Satya Nadella</a>
<a href="#">Alphabet</a>	<a href="#">Mountain View, CA</a>	135,301	<a href="#">Sundar Pichai</a>
<a href="#">Amazon</a>	<a href="#">Seattle, WA</a>	575,700	<a href="#">Jeff Bezos</a>
<a href="#">Disney</a>	<a href="#">Burbank, CA</a>	201,000	<a href="#">Bob Chapek</a>

Nope, not even this. This is a different use case, which we will talk about later.

However, the real question is, doesn't the user already know? Is there even a problem to fix in the first place? In reality, this is a non-issue. The user can simply click the row and navigate to the detail page. Tapping anywhere inside a

The funny thing is, I would tell that same stakeholder to "*whip out your phone and try and do this task yourself, on your mail app, right now.*" The stakeholder then executes the task flawlessly. I continue: "*How did you know you can tap the row of your mail list view?*" It's then when they realize, they've always been using it correctly, without actively thinking about it and without the need to have something in the UI explicitly tell them that they can "*click the row to go to the detail view of this item*" type of thing. It's also when some still are convinced there should be a call to action nonetheless and won't budge, regardless of their revelation... But that's a different kind of discussion.



iOS does indicate a forward navigational motion with the *right pointing chevrons*. However, there is no explicit "click here" indication to be seen. Regardless, users find navigating to the detail page of an email easy to find. (image credit [iMore](#))

Company	Location	Employees	CEO
Apple	<a href="#">Cupertino, CA</a>	147,000	<a href="#">Tim Cook</a>
Microsoft	<a href="#">Redmond, WA</a>	166,475	<a href="#">Satya Nadella</a>
Alphabet	<a href="#">Mountain View, CA</a>	135,301	<a href="#">Sundar Pichai</a>
Amazon	<a href="#">Seattle, WA</a>	575,700	<a href="#">Jeff Bezos</a>
Disney	<a href="#">Burbank, CA</a>	201,000	<a href="#">Bob Chapek</a>

For our table example, the user would be confirmed by a hover state. One can discuss if this has sufficient discoverability and whether a chevron such as used in iOS would be necessary. iOS and MacOS have some usability issues, but that's for another discussion.

## Don't make them think

It's my experience that often, when stakeholders start to actively think about things, they tend to question heuristics that might actually have been proven and implemented correctly long before. One can speculate why that is. Maybe it's a matter of wanting to prove their role matters, maybe they're just overthinking it, maybe it's a cognitive fallacy, or maybe even plain unfamiliarity with the matter. In any way, I typically take these types of feedback with a grain of salt.

*A quick pro tip: to prevent your UI design to be Frankenstein'ed into pieces by random stakeholder input, try using the hairy arm method! But be aware of how and when to use it! (by [Chris Kernaghan](#))*

That said, does this disqualify the "*button, what else?!*" responses completely? No it does not, as I think it is quite illustrative as to how most look at UI. It's the UI designer's task to make sure normal users don't have to think about these kind of subjects and just experience an optimal interface. Even if it's subconsciously; maybe even *especially!*

**"Cancel is an action" & "Links are about going to pages"**

Person on the internet

**Cancel is an action and therefore a button.**

Person on the internet

**Links are about going to pages.**

Me

**Cancelling is not an action and links are not solely about "going somewhere".**

This sounds an awful lot like you either disagree with the original premise (which is of course a completely fair thing to do), or you haven't really read or understood the points laid out in the previous article. There I state and explain why cancelling is *not* an action and that links are *not* solely about "going somewhere", but about navigating between states. **Let's revisit and further explain this into more detail**, as I believe this is actually the crux of the previous story.

To really get a grasp of the idea, we have to do yet another classic little dive into the history of the web...

## Web 1.0 is about content and pages

Back in the day of the old *Web 1.0*-era (when we didn't call it *Web 1.0*, but just the *world wide web*, and where we wouldn't stop saying "http://www" in front of every single URL), websites were basically just documents of text (headers, paragraphs) and images (like Word documents, but online). Keep in mind though, that the concept of "online" was still in its infancy. Tim Berners-Lee, the father of the internet, came up with a way to structure those documents' contents semantically, which would allow programs that read those documents (browsers) to display them correctly. Those documents then could consist of distinct pages, all referenced by hyperlinks pointing to each other and themselves. A hyperlink typically explained to the user what they would navigate to when clicking, and



This [Great Calvin and Hobbes Ring](#) site is owned by [Your Name Here](#).

Want to join The [Great Calvin and Hobbes Ring](#)?

[[Skip Prev](#)] [[Prev](#)] [[Next](#)] [[Skip Next](#)] [[Random](#)] [[Next 5](#)] [[List Sites](#)]

The classic example

Eventually, those web documents, became more and more complex in what they had to offer content-wise. Amazon.com for instance, grew into a huge website, full of pages with lists of books and book detail pages and pages to pay for those books and what not. We can imagine bringing the user from one page with a certain "state" (e.g. a list of popular books this year:

`amazon.com/books/popular/year` ) to another page with another "state" (e.g. a list of popular books this week: `amazon.com/books/popular/week` ). In the end, all we did was simply navigate between different pages with different URLs, which would reflect the contents (state) of the website. When traversing through the URLs, your browser would keep track in the browser's history. Clicking your browser's back (or forward) button would allow a user to backtrack in either direction. In a sense, Web 1.0 was about links.

The screenshot shows the classic Yahoo! homepage. At the top, there's a navigation bar with icons for 'New' (a baby), 'Cool' (a smiley face), and 'Today's News' (a newspaper). The main title 'YAHOO!' is in its signature red, bubbly font. Below it, there are several promotional banners: 'Win a Jeep from ichat' (with a small image of a jeep), 'Kingston' (with a small image of a Kingston logo), 'Click here to enter contest.', 'Win Free Music for Life' (with a small image of a person holding a guitar), and 'Yellow Pages - People Search - Maps - Classifieds - Personals - Chat - Email'. A search bar with the placeholder 'Search' and a 'options' link is also present. Below the search bar is a horizontal menu with links like 'Yellow Pages', 'People Search', 'Maps', 'Classifieds', 'Personals', 'Chat', 'Email', 'My Yahoo!', 'News', 'Sports', 'Weather', and 'Stock Quotes'.

- **Arts and Humanities**  
Architecture, Photography, Literature...
- **Business and Economy [Xtra!]**  
Companies, Investing, Employment...
- **Computers and Internet [Xtra!]**  
Internet, WWW, Software, Multimedia...
- **Education**  
Universities, K-12, College Entrance...
- **Entertainment [Xtra!]**  
Cool Links, Movies, Music, Humor...
- **Government**  
Military, Politics [Xtra!], Law, Taxes...
- **Health [Xtra!]**  
Medicine, Drugs, Diseases, Fitness...
- **News and Media [Xtra!]**  
Current Events, Magazines, TV, Newspapers...
- **Recreation and Sports [Xtra!]**  
Sports, Games, Travel, Autos, Outdoors...
- **Reference**  
Libraries, Dictionaries, Phone Numbers...
- **Regional**  
Countries, Regions, U.S. States...
- **Science**  
CS, Biology, Astronomy, Engineering...
- **Social Science**  
Anthropology, Sociology, Economics...
- **Society and Culture**  
People, Environment, Religion...

---

[Yahooligans! for Kids](#) - [Beatrice's Guide for Women](#) - [Yahoo! Internet Life](#)  
[What's New](#) - [Weekly Picks](#) - [Today's Web Events](#)  
[Visa Shopping Guide](#) - [Yahoo! Store](#)

[World Yahoos](#) [Australia & NZ](#) - [Canada](#) - [France](#) - [Germany](#) - [Japan](#) - [Korea](#) - [UK & Ireland](#)  
[Yahoo! Metros](#) [Atlanta](#) - [Austin](#) - [Boston](#) - [Chicago](#) - [Dallas / Fort Worth](#) - [Los Angeles](#)  
[Get Local](#) [Miami](#) - [Minneapolis / St. Paul](#) - [New York](#) - [S.F. Bay](#) - [Seattle](#) - [Wash D.C.](#)

Smart Shopping with

---

[How to Suggest a Site](#) - [Company Information](#) - [Contributors](#) - [Yahoo! to Go](#)

Links everywhere! (image credit <https://www.versionmuseum.com/>)

## Web 2.0 is about interactions and collaboration

It's when Web 2.0 happened, that we started to see more complexity in terms of interactions. **Web 2.0 was about sharing and creating.** JavaScript proved to be a great tool to update parts of a page, instead of loading an entire new page, which would allow for those sharing and creation interactions.

---

**Wikipedia** A Web 2.0 website allows users to interact and collaborate with each other through social media dialogue as creators of user-generated

manner.



Who still remembers?

We don't realize it now anymore, but back then, that was revolutionary! In the years to come, more and more complex interactions were invented, which overshadowed the passive Web 1.0 world fast, by the inevitable rise of more complex websites, or so adeptly named, *web applications*, which would allow content creation and collaboration in the browser. Writely (now Google docs), Youtube, Flickr, Google Maps, Netvibes and Twitter were all born in that era. Websites no longer were "just simple documents" only. But it also introduced challenges. How do we handle navigation? How do we reflect pages and states? Does this still work?

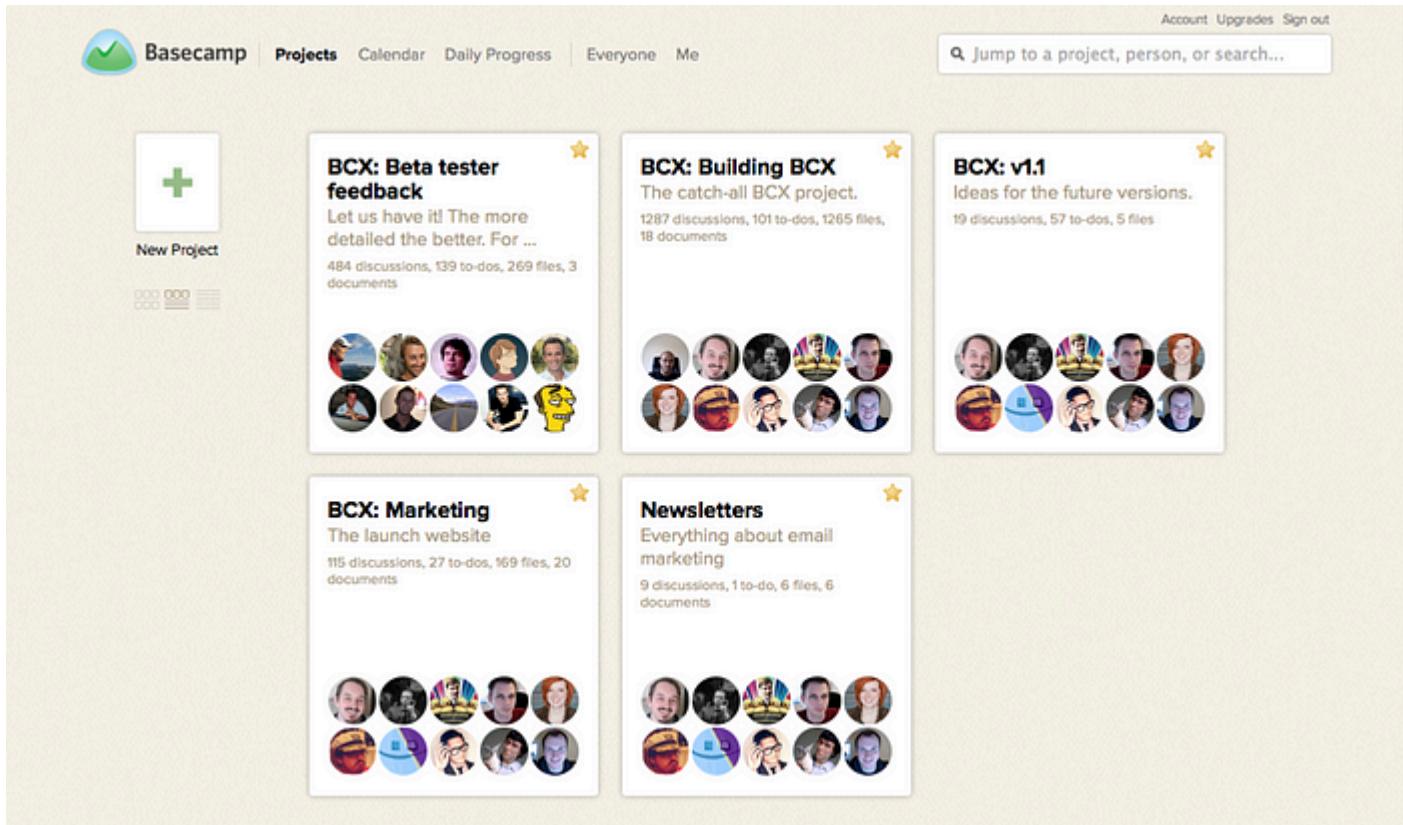


Included: the famous web 2.0 gradient

## HTML5 State Management

Basecamp was a good example of a web application where the HTML5 History API was put to good use. When navigating between states or pages, only *parts* of the navigated page were loaded. No need to reload an entire menu, logo or footer, when all that changes is the body only. The URL subsequently automatically updated to reflect this page (state) change. [basecamp.com/projects](http://basecamp.com/projects) → [basecamp.com/projects/acme](http://basecamp.com/projects/acme) — This resulted in a fast user experience, while still being able to browse and navigate as usual. On top of that, those URLs were

online writing tool such as Writely, where a version history was reflected in the URL. Or for an online map service, such as Google Maps, where location, zoom and scale was reflected in the URL and could be shared and bookmarked.



Basecamp version 2

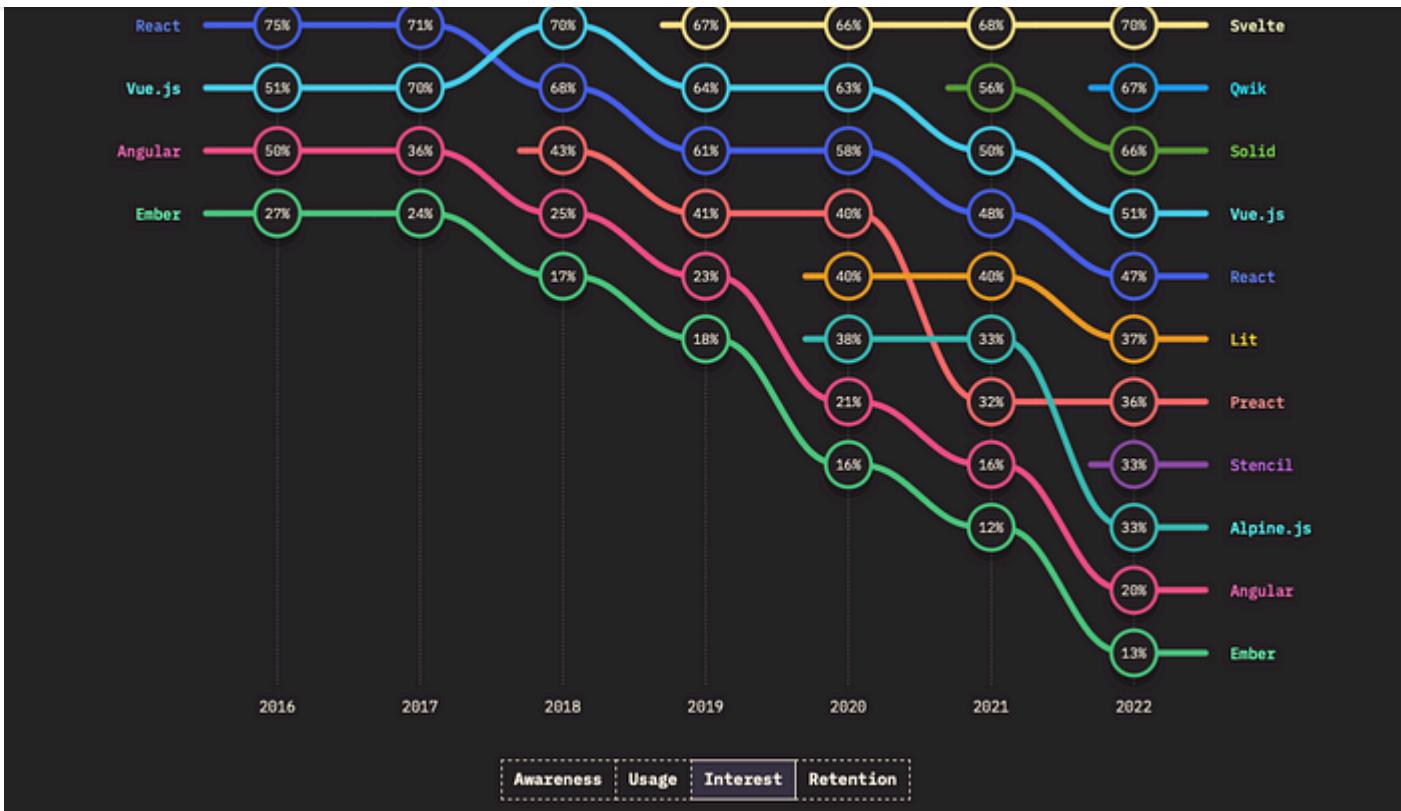
Up until this point, these implementations were all custom build. It became painstakingly clear that there was a need for standardized JavaScript behavior and implementation, in order to do these fancy feats somewhat efficiently and effectively. This gave birth to tools such as [Scrip.taculo.us](#), [Mootools](#) and later [jQuery](#), assisting front-end developers with dealing with the cross-browser quirks of JavaScript (or just [not dealing with vanilla JavaScript at all](#)).



script.aculo.us — Fancy URLs were a thing back then

## Modern Web Development

This foreshadowed the ultimate rise of modern web development: the era of Knockout, Ember and the old AngularJS. These JavaScript frameworks did the heavy lifting for developers. Instead of dealing with hybrid situations such as the one Basecamp created, or creating your own implementation with jQuery (I'm guilty of doing that), these MVVM frameworks pretty much relocated the back-end logic of old multi-page websites (MPAs) to the front-end, allowing for smart data-binding, quick partial navigation (with or without the HTML5 History API) and other tricks that static pages couldn't do, all in a single-page application (SPA)! These days, we have the more moderns version of Angular, Vue, React, Svelte, and many more.



From [stateofjs](#)

Despite taking place, technically, all in one single page — in the end — we are still looking at a multitude of pages that are interlinked by references, reflected by a URL. This has not fundamentally changed. On the contrary, it ensures we are tapping into what we have come to know and appreciate of what is the power of the web. We retain a functional understanding and working of the web as we know it, so that we don't break the web and empower the user.

**developer.mozilla.org: Don't break the web** The idea is that any new web technology that is introduced should be backwards compatible with what went before it (i.e. old web sites will still continue to work), and forwards compatible (future technologies in turn will be compatible with what we currently have).

**Smashing Magazine: Designing A Better Back Button** With it comes the expectation that the "Back" button will bring a user to the previous "page," even though, technically, it might not really be a separate page.

or breaking the user's expectations of how the browser back button should behave can be dire. During our usability tests, it has been the direct cause of abandonment, with users leaving test sites with much swearing and cursing (even from the calmer test users).

## Progressive enhancement for the win

But, what does all of this have to do with the idea of "links are navigation"? Good question. Let's explore some examples.

Person on the internet

**But, what does all of this have to do with the idea of "links are navigation"?**

Chris Coyier put it quite nicely in his blog:

If there is any kind of href you could put on that link that makes sense, by all means, use an anchor (link). Even if you override that behavior with JavaScript. That's progressive enhancement at its finest.

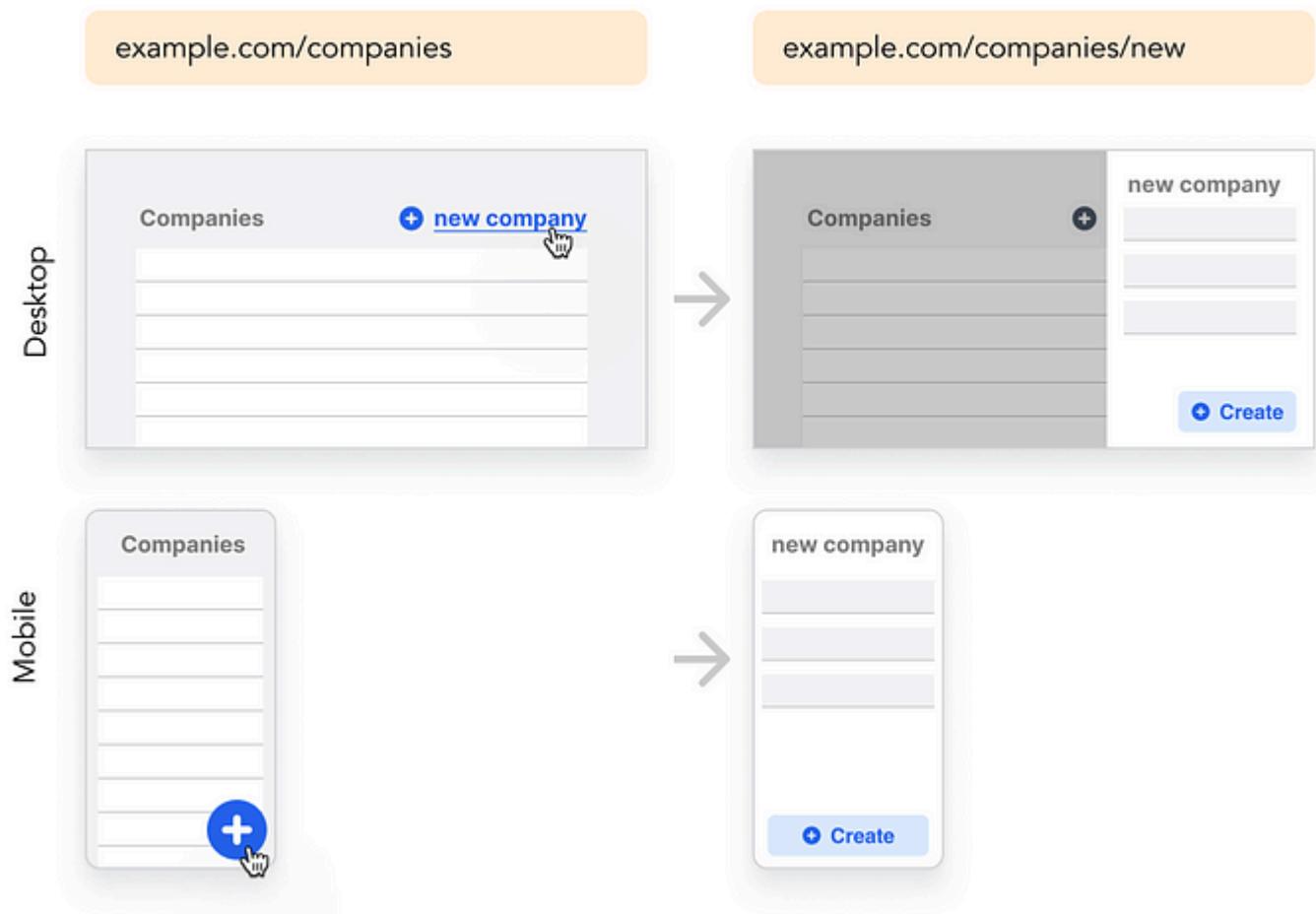
Chris then continues with these examples:

- A search button normally triggers a type-ahead ajax-search thingy – but the href could just point to a /search/ page.
- A publish button triggers the next stage of publishing something a user built – but the href could just point to a /publish/ page.
- A thumbnail button opens a lightbox with a larger version – but the href could just point to the URL of that larger version.

Here we clearly see how links such as these are **very real points to a reference**. It's navigation, as the user is doing a UI state change, *despite* the fact that a search widget, or a publish side panel, or a light box overlay may *feel* like the user still

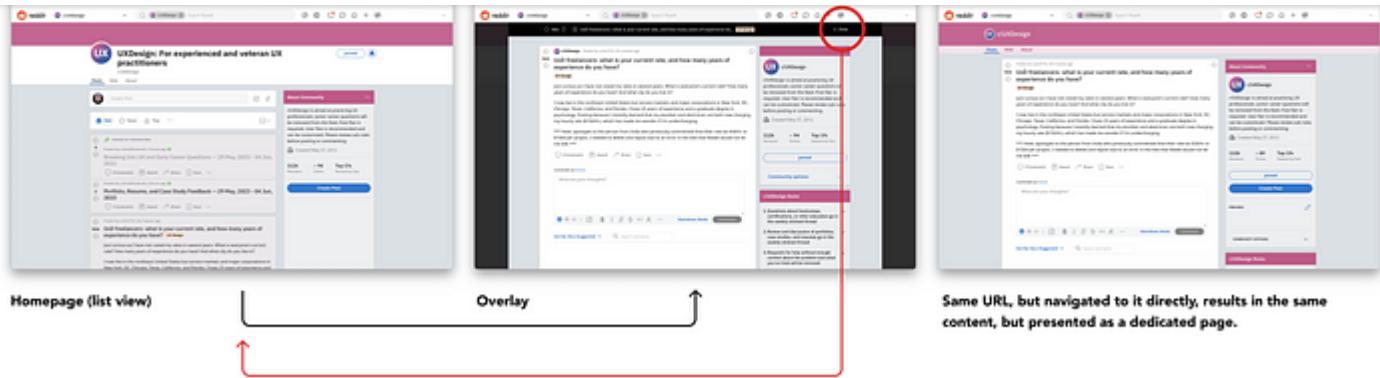
## Smashing Magazine: Always Close Large Overlays With The "Back" Button

And it's not an action as there is no underlying data manipulation. In fact, there is nothing stopping us from displaying the light box overlay as used in Chris Coyier's example as a full page when navigating directly to it, instead of as displaying it as an overlay when navigated to it from a list view. The same goes for any (modal) view for that matter.



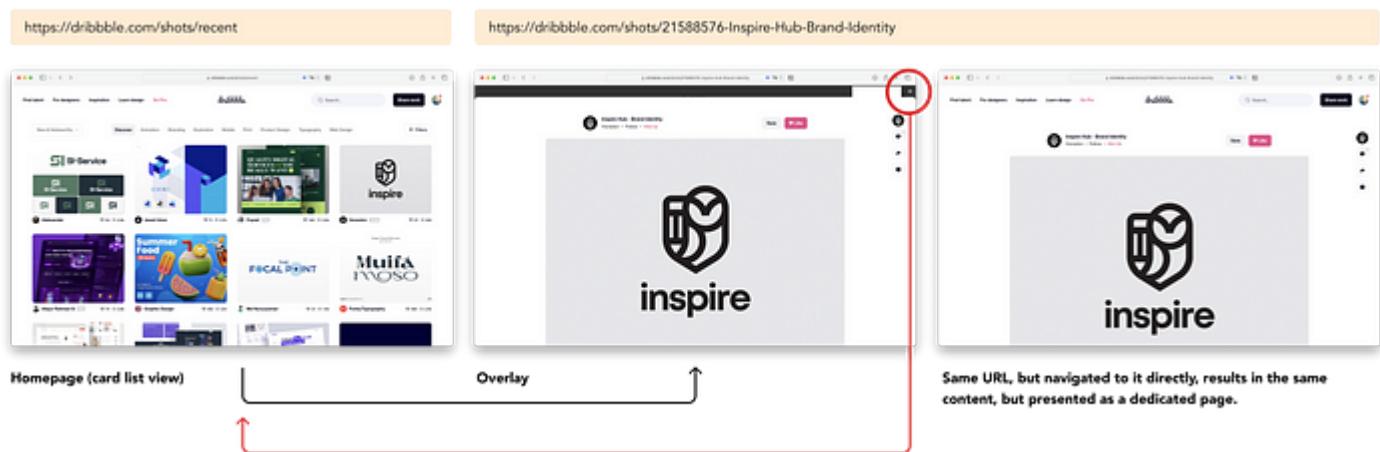
Or, when going to the exact same page on desktop or mobile, a different view can be displayed; side panel vs. full page. Same URL, different view.

**Reddit uses this pattern extensively on their new Reddit 2.0.** When on the list of stories (the reddit homepage, or a subreddit's homepage), a user can click on a story, which then opens as an overlay on top of the list. The URL changes to reflect to state change. But when navigating directly to that URL, the overlay becomes a full page.



Clicking on a story, navigates to an "overlay view", showing the user that story. This overlay can be closed by clicking outside of it, or by clicking on the "Close" button-ish (booo!) on the top right of the overlay. However, navigating to the URL of the story directly, only shows the story as a full page. No overlay. Notice the URL: it's exactly the same as the overlay page.

Dribbble does the same trick. Both examples clearly demonstrate that having an 'overlay' view is irrelevant to the fact we're still dealing with navigation from one URL to another.



Despite the fact those page changes (or state changes if you will) are done through the means of JavaScript (and while technically under the hood these are single pages), **we are still doing basic navigation**. For a user, there is *no difference* between the old Web 1.0-variant of navigation, and the modern one build with an MVVM framework.

## Conclusion

In both cases, we are presenting the user with links to navigate between different states (views) of the website. These links point to URLs which can be bookmarked and shared if desired. As the cancel (or close) option in a dialogue view is similar

## Other remarks

Persons on the internet



Let's build on what we've explored and learned so far. I'll spotlight some individual remarks (no hate, only discourse!) to see if we can agree or disagree and why.

### "Cancel is rolling back, so it's a button"

Person on the internet

**Cancel is rolling back, so it's a button**

Me

**I think this is almost correct, except for the fact that the user would *not* have committed any actions just yet.**

— I think this is almost correct, except for the fact that the user would *not* have committed any actions *just yet*. If the user would have gone into a next step, having applied data manipulation, then a "cancel" would indeed be a button, as it conveys rolling back of data as an action. (I would also prefer not calling it "cancel" then, but "roll back").

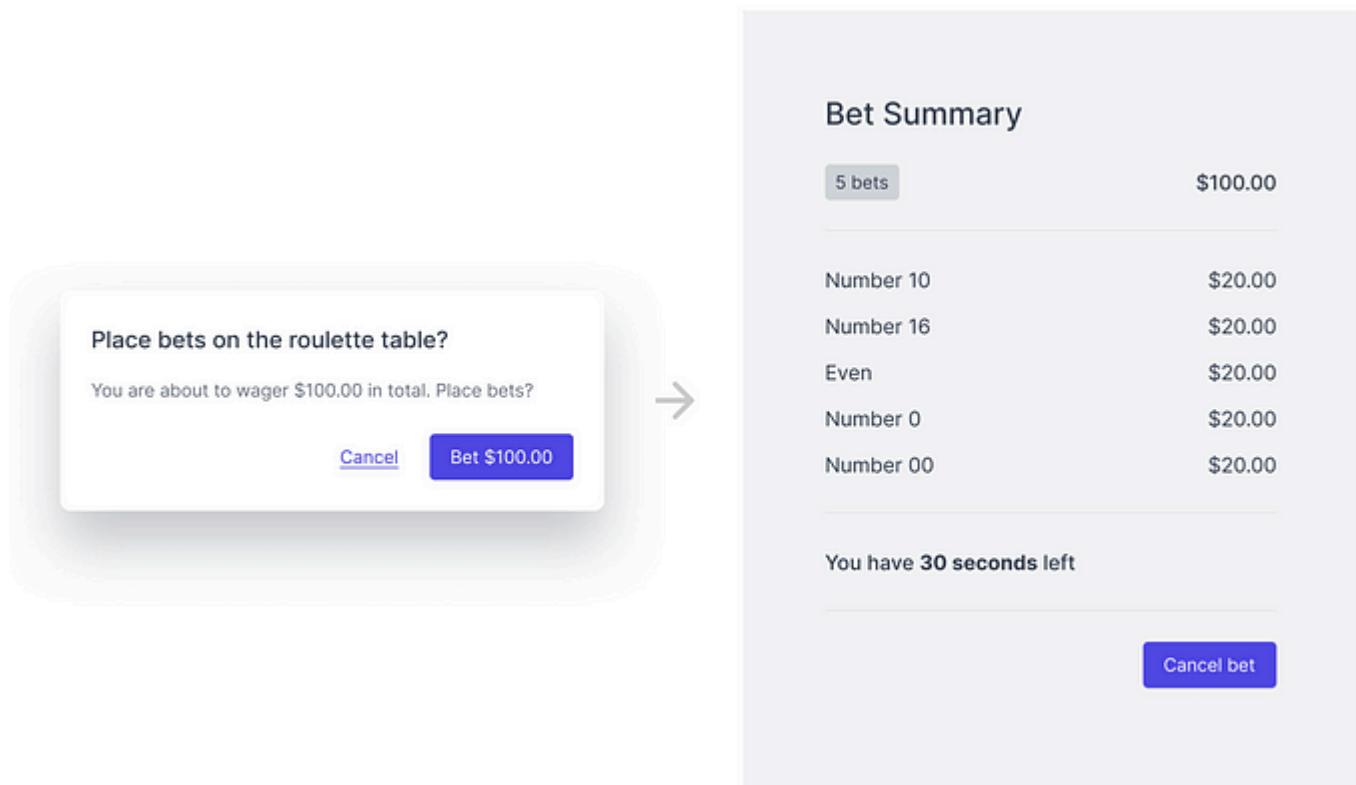


Maker 🤖 Software Engineer 🧩 Software Architect 🏠 Legacy Syst...

Cancel should be a button and it should actually roll back any changes that the other button would commit.

I have to disagree

And again, as stated before, actions *can* have consequential navigation attached to them. Clicking a "save" button would save data but also possibly navigate the user to a new page where they can see the newly created data. A user clicking the "roll back" button would also apply data manipulation (rolling back of data) and then be brought to a new state of the application, where the new data is seen. Consider this example of a roulette betting dialogue:



When placing bets, the user is prompted to confirm their bets. They can still click cancel, which would allow them to not continue. It does not do any data manipulation and **thus is merely navigation**, hence a link.

Once the bet's been confirmed, data *has* been modified (i.e. the bet exists) and the user is presented with an overview, allowing them to cancel the bet. In this case, cancel is actually an action, as it is an **undo of the already applied data**; it

earlier.



Adam Robertson • 3rd+

An award-winning UX/UI and product designer. Also a senior-level f...

1w \*\*\*

It's all relative to the context of the UI. If "cancel" merely dismisses a modal, it's rather moot. If it does more, like rolls back changes, it technically should be a button. On how it's styled is again, rather moot. Although styling it in a way to minimize cancels would be considered a dark pattern.

Exactly!

## "Links are for linking only!"

Person on the internet

**Links are for linking only!**

Me

**Kind of, yes. But "linking" means a bit more than you might expect.**

— A link is a reference between two URLs of your web page or application. These URLs reflect different "states", such as

- showing a list of resources ( `example.com/fruits` )
- showing the same page in a different format ( `example.com/fruits?view=cards` )
- having that list filtered ( `example.com/fruits?color=red` )
- showing a resource detail page ( `example.com/fruits/apple` )

In all of these cases, the user can interact with your application through the means of clicking those links (which would present the user different views of



Ismail Ahmad • 3rd+  
UX Designer

1w \*\*\*

A link signifies a LINK... Canceling is not linking, it is an action, and important one for that matter. Making it a button will always be the best approach.

Cancelling certainly isn't data manipulation. It is however, the user navigating away from the dialogue.

Cancel certainly is *not* data alteration, but merely navigating from one state to another state (view) of the UI, without changing the data. In fact, it's for the user to **explicitly not to choose to do data manipulation**. The user chooses to *navigate* away. Hence, it's a link.

Another interesting discussion I got into, that kind of is the cumulation of many of the points that I tackle here, was the following:



Maya Mircheva • 3rd+  
Frontend Developer at Avanade

1w (edited) \*\*\*

The article suggests that nearly all actions in an app require navigation (and should use links) instead of data manipulation or "real" action that should warrant a button.

I tend to disagree with that because in the example of a modal above the cancel button doesn't redirect back (we live in the age of SPAs after all) but simply toggles off the visibility of the modal. In code this is achieved by a function attached to the click event so definitely an action. Even for the case of links, as previous comments suggested, it is still best to style them as button if they are important enough and need to draw the user's attention.

Besides, they offer a larger clickable area than a simple text with underline which is always better for accessibility. I would argue for links style (blue text with underline) only in the case of inline links when they are mixed with paragraph text. Even main navigation could be links with no underline since users already know that the navigation is on top and it is clickable.

Let's expand on this.

Here, their message basically is "the article suggests that nearly all actions in the app require navigation (and should use links)." — Yes. That's exactly what I

of it being JavaScript functionality, closing a modal view is an action." — I disagree. By that same token, pretty much *everything* would be an action. Sure, SPAs are indeed wired with JavaScript. But that doesn't mean all things are all of a sudden an "action", just because you do event handling in JavaScript behind the scenes. Actually, it's quite the opposite. It's *regardless* of the fact that we use JavaScript to apply functionality to elements, that most of these interactions are links nonetheless. And, it's *especially* because of SPAs, that we tend to forget about what it really is we're actually doing with our web interfaces. Remember, don't break the web!

To hammer it home: navigating back and forth in your application, is nothing more than just **going back and forth in your browser history**, regardless of how it is presented visually (e.g. a modal view, overlay, side panel, dedicated page), regardless of how it's been build. So, going from one state where there is no overlay, to one where there is, is just navigation and nothing more, and should be reflected by (1) a link that navigates the user to that new state, and (2) a URL change. (e.g. `example.com/fruits` → `example.com/fruits/new`) This would then result in the user being able to use the browser's back button to 'return' to the previous state without an overlay.

## [Smashing Magazine: Always Close Large Overlays With The "Back" Button](#)

Lastly they mention that "links can be styled as a button anyway" and that doing so also "offers a larger clickable area." — I disagree. But styling and Fitts's Law, I will address further down this article.

## "It's a dark pattern"

Person on the internet

**It's a dark pattern**

Me

**It's not. I'll explain.**

- It's not. I'll explain. Dark patterns (or nudge techniques) are interface techniques to trick users into doing something they would otherwise not want to be doing.

Deceptive design says: Deceptive patterns (also known as "dark patterns") are tricks used in websites and apps that make you do things that you didn't mean to, like buying or signing up for something. For example: Trick wording, Sneaking, Obstruction.

Harry Brignull (*who came up with the term*) says: A user interface that has been carefully crafted to trick users into doing things..they are not mistakes, they are carefully crafted with a solid understanding of human psychology, and they do not have the user's interests in mind.

So a few takeaways here: dark patterns are about trying to move users into a direction they might not want to go and without having the user's interest in mind, doing so while being fully aware of human psychology.

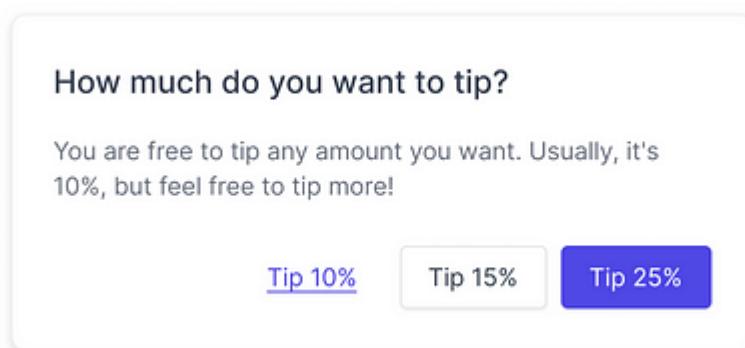
When applying cancel as a link, it's absolutely *not* about nudging the user in a direction they would otherwise not want to be going. On the contrary; it presents the user a semantically congruent way of navigation, allowing them to make a conscious and thought-out decision. You're, in fact, *helping* them! When ignoring the distinction between the two different types of interactions (buttons vs. links), the user is less confident about what to expect. This would actually be ignoring their interest (i.e. being clear about the intent).

The image shows two identical-looking tables side-by-side, each listing five video games with their platform and price. The first table on the left contains four standard 'Buy now' buttons, while the second table on the right contains four 'Buy now' links.

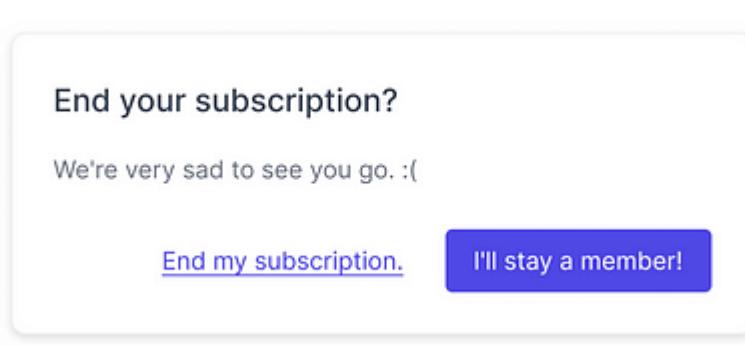
Game	Platform	Price	Action
Super Mario Odyssey	<a href="#">Nintendo Switch</a>	\$59.99	<a href="#">Buy now</a>
Breath of the Wild	<a href="#">Nintendo Switch</a>	\$59.99	<a href="#">Buy now</a>
Tears of the Kingdom	<a href="#">Nintendo Switch</a>	\$69.99	<a href="#">Buy now</a>
Yoshi's Crafted World	<a href="#">Nintendo Switch</a>	\$59.99	<a href="#">Buy now</a>
New Super Mario Bros. U Deluxe	<a href="#">Nintendo Switch</a>	\$49.99	<a href="#">Buy now</a>

The first tabel has 'buy now' buttons. This would imply data manipulation. A user can expect a buying action to take into effect immediately when clicking the button. The second table has 'buy now' links. This would imply navigating to a view (side panel, page, etc) where the user can proceed with the buying journey. They would not have done any buying actions yet after clicking the link, as it's purely navigation.

These examples would correctly be called dark patterns or nudge techniques:



We're nudging the user into tipping 25% by abusing different UI types.



Here, we're nudging the user into staying a member by abusing different UI types.



## WE'RE SORRY THAT YOU'RE THINKING OF UNSUBSCRIBING...

Email Address

BEFORE YOU GO – WOULD YOU LIKE TO HEAR FROM US LESS OFTEN INSTEAD?



SEND ME FEWER EMAILS

---

STILL WANT TO UNSUBSCRIBE FROM NEWS ABOUT OFFERS, COLLECTION LAUNCHES AND EXCITING NEW PRODUCTS?

We're sad to see you go but feel free to come back anytime.

UNSUBSCRIBE FROM ALL

Love the shop. Hate the dark pattern.

"It depends"

Person on the internet  
It depends

Me  
It always does, doesn't it?

— It always does, doesn't it? Question is, it depends on what exactly? I think it depends on what I mentioned earlier: is it altering data (an action)? Then it's a

## "I just prefer the look of a button"

Person on the internet

**I just prefer the look of a button**

Me

**I guess that's subjective. However, I find a functional and predictable interface visually more pleasing than one that's been crafted purely on aesthetics.**

— I guess that's subjective. Of course, creating visually pleasing UI, definitely contributes to an increase in usability. However, applying predictable patterns is just as important to ensure usability, if not more. This actually stresses the fact of why it's so important to send a clear message with your UI design. If "cancel" would be a button, it would incorrectly indicate a different message to the user, than if it were a link. And to be honest, I find a functional and predictable interface visually more pleasing than one that's been crafted purely on aesthetics. Which brings us to:

**"It's about styling anyway, so it doesn't matter as long as you style it so it's clear" & "I don't like underlined links"**

Person on the internet

**It's about styling anyway, so it doesn't matter as long as you style it so it's clear**

Person on the internet

**I don't like underlined links**

Me

**Styling a link as a button or a button as a link (or any other UI breaking style) is considered bad practice. We are still talking semantics here and by presenting the user UI in a predictable fashion, they understand what they're dealing with.**

- How a UI designer styles their hyperlinks and buttons is of course to their discretion (blue, green, underlined, background color, etc...). However, **styling a link as a button or a button as a link** (or any other UI breaking style) is considered bad practice. We are still talking semantics here and by presenting the user UI in a predictable fashion, they understand what they're dealing with, thus "making it clear". By the way, there *is* a preferred way of styling links.

*Marcy Sutton says: Push back on design to make links look like links and buttons look like buttons. Removing the ambiguity makes it easier for developers to code more accessibly and better meet user expectations.*

*furbo.org Design is not how it looks, it's how it works.*



**Lizz Brazen** (She/Her/Ms) • 3rd+

Experience Design Leader | User Experience is Brand Experience

5d \*\*\*

Never style your hyperlinks as buttons or your buttons as hyperlinks. ✨ NEVER ✨

Exactly!

## "But link buttons and ghost buttons exists!"

Person on the internet

**But link buttons and ghost buttons exists!**

Me

**It's true they exist, but that doesn't mean they're used correctly, or should be used at all.**

— It's true they exist, but that doesn't mean they're used correctly, or should be used at all. Or as Olivier likes to spice things up:



Olivier Mattelart • 2nd  
UX Designer at ADESA Europe

1w \*\*\*

You forgot LINK buttons, as they exist (see Material Design), just to put more spice into this guerilla of specialists...

I choose to ignore them :P

Ah yes, Material UI. The bane of our UI existence. Keep in mind that Material UI doesn't always have it right. A good example of where they made a bad usability mistake is their (now deprecated) input field UI. [Smashing Magazine wrote a great article about this.](#)

Anyway, to actually go into discussing link buttons or ghost buttons: They started at around 2014 as a flat UI trend. While they might appear as a link in some cases, **borderless buttons are not the same as a big link** with a decent clickable line-height. They are two semantically distinct elements and thus have different meanings to the *browser*, the *user* and their *accessibility*. Nothing is stopping us from making a link more clickable, if that's your concern (which is a valid one), so why use a link-styled button anyway?

Ever noticed that you're perfectly capable of tapping a link inside a paragraph on your smartphone? Mobile operating systems have a smart way to go about predicting what your intentions are when touching the UI with your big fingers.

Obfuscating the fact a button is a button (e.g by removing its borders, or presenting it like a link) **is a bad idea to begin with**, as we've explored many times before. So, yes, link buttons or ghost buttons exist. But it's quite difficult to find real use cases for them other than to dive into dark patterns or resulting into bad performance or misusing them entirely.

## "Using buttons is more consistent"

Person on the internet

**Using buttons is more consistent**

Me

**The thing about consistency is that it's often misused or misinterpreted. While consistency can be a good thing, it can also be a bad one, if not applied correctly.**

— The thing about *consistency* is that it's often misused or misinterpreted. While *consistency* **can** be a good thing, it can also be a **bad one**, if not applied correctly.

In the end, *consistency* in itself is not the goal. Usability is the goal, and while *consistency* *can* attribute to that, if *consistency* is forced just for the sake of being consistent, it can actually hurt your usability.



80% of the time, consistency always works!

**Consistency often becomes an idesfix**, which due to the lack of understanding your user interactions, **removes the need to grasp the complexity of your solutions**. It's *lazy design* and makes for a sub par interaction.

Consistency is *not* about making everything look the same. It's about making sure different things behave in their respectively different but distinct (and predictable!) ways. This ensures users understand what to expect from a certain interaction. Or as Mark Parnell so nicely states: "One of the more insidious mistruths has been the need to always be consistent with your design work." I agree.

Consistency should empower the user; helping them understand what to expect from the UI. By making everything *seem* the same, it *feels* the same and loses its distinct language to convey its functionality to the user.

Carlos Yllobre says: Design for coherence, not consistency. Consistency and coherence might seem like very similar concepts, but they are fundamentally different. Consistency's sole purpose is to make the elements of a system uniform. It is about how similar the parts look and feel throughout the entire system. On the other hand, coherence aims for clarity. It is about how well something is understood and if it makes sense. If a system is coherent, it will be clear and easy to understand.

Thinking of consistency as the primary goal of your product or design system can result in clean libraries, tidy components and many things feeling and looking the same, but this doesn't equal a good experience for your users.

## Freedium

what I'm designing consistent with other things we've designed (or others have designed)?" is the wrong question to ask. Instead, the right question is, "Will the user's current knowledge help them understand how to use what I'm designing?"

Being consistent for the sake of being consistent would diminish our users' ability to discern different elements with different functions and purposes. However, being consistent in your distinctions, empowers your users; they will be able to predict what to expect when interacting with your UI.

**Adam Silver** says: "consistency is not about making different things the same. It's about making the same things the same."

I think UX Planet explains this quite well too:

Consistency within our interface builds a constraint. Although constraints are not bad, they can divert us from achieving our goal just to follow the rule.

Sometimes, consistency can lead us to design non-usuable interfaces just to checklist that our designs are consistent. Example: Following consistency in a place where it makes it more difficult for our users to achieve their goals.

Consistency can be a strong argument to justify our non-usuable designs. It's safer and easier to design consistent experiences than designing usable ones.

And:

We need to ask ourselves every time we say "Consistency", do we follow this principle for the sake of being consistent? Will consistency here achieve user goals? Do we have an easier and usable way to do so?

## "Use whatever your design system uses"

Person on the internet

**Use whatever your design system uses**

Me

**You always have the choice to deviate when needed. Challenging a design system should an option if there's a reason to do so.**

— I agree that once a design system is in place, it's best to use your system of choice as much as possible for your UI designs. However, in the end, you always have the choice to deviate when needed. On top of that, challenging said design system should always be an option if there's a reason to do so. Maybe, the system's designer has fallen into patterns that are actually sub-optimal and need updating. Maybe they haven't thought about something you have. Besides, not all design systems are strict or even explicit when dealing with patterns or templates, and leave the UI designer free to interpret it as they please.

## "Buttons are default and you see them everywhere"

Person on the internet

**Buttons are default and you see them everywhere**

Me

**Just because others do it, doesn't mean you have to as well.**

of that, just because others do it, doesn't mean you have to as well. However, I want to point out that common ground patterns empower users due to familiarity (Jakob's Law). It's also exactly why I want to challenge the pattern of "*always buttons*" to prevent it from becoming the default in the future.

Keep in mind that some time ago, we went really quickly to a "hamburger menu" pattern in mobile and responsive designs, while many objectively concluded that this is bad practice (great article [Michael J. Fordham!](#)). Without challenging ideas, we don't move forward or are able to correct ourselves.

#### Nielsen Normal Group on hamburger menus:

"Like a cheap fast food chain, it got designers addicted to its convenience, and now serves millions each day, both on mobile devices and on desktops."

'Convenience' definitely comes into play in my opinion. When we don't want to think about these types of subjects, it's much simpler to just ignore them and apply cookie-cutter designs. That said, just to give you an idea of how prevalent cancel (or similar) links still are, here a quick mood board:

**"It's a binary choice so it should be of equal weight / presentation" & "a simple yes / no suffices" & "primary, secondary, tertiary styles apply here"**

Person on the internet

**It's a binary choice so it should be of equal weight / presentation**

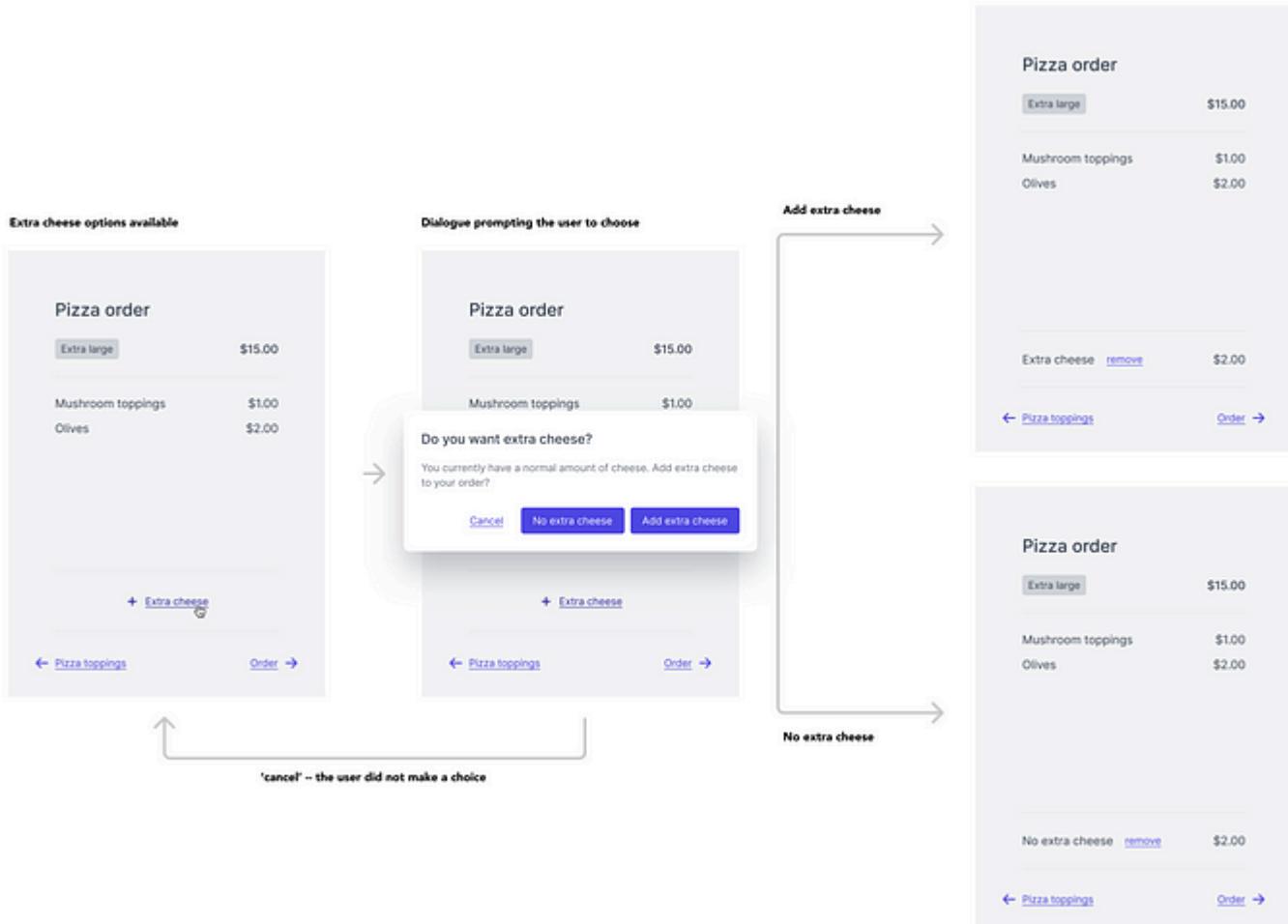
Person on the internet

**A simple yes / no suffices**

Me

**It's actually not a binary choice.**

- It's actually *not* a binary choice. Let's take a look at the following example of ordering a pizza. The user is presented with an "extra cheese" option. In the first view, they have three options: (1) "Extra cheese", (2) "Order", which would proceed the user to the order view. (3) "Pizza toppings", which would navigate the user back to the pizza toppings view.



adjust the extra cheese option there.

When clicking "Extra cheese", a modal view (prompt) is shown. The user then can make a decision on whether they want extra cheese, or don't want extra cheese. Both options are a decision on their extra cheese, and thus both options are a button (i.e. data is manipulated and the extra cheese flag is set to either true or false). They are also opposing options.

When clicking "cancel", the user opts to *not* make a decision and thus leaves the question whether they want extra cheese or not, open. They will therefore return (navigate back) to the previous view, where adding extra cheese is still available as an option. Hence why "cancel" is a link. But also why "cancel" is *not* an opposing option. In this context, "cancel" is not about the user deciding they don't want extra cheese. They decide to leave that option open.



Varduhi Harutyunyan • 3rd+  
UX designer

1w \*\*\*

The main problem and arguing thing here is the confusing UX Writing. If the cancellation is just to close a pop-up it could be a link (but it must have a "Close" label. But if it is an opposite for the Save button, for example, it MUST be a button and its label must be "Discard Changes" or just "Discard" because it is a decision and action.

Indeed.

If the "cancel" is an **opposing action, then it's a button.** (e.g. "I don't want extra cheese" vs. "I want extra cheese") — If it's just "closing" (in other words, the user does not make a choice at all, but wants to abort the decision), then it's a link. In the example of Varduhi, the "Discard Changes" as suggested, would only be a button if it's actually undoing already done data manipulation. One could actually see an interaction like this in the final state of the pizza order example:

Pizza order

Extra large	\$15.00
<hr/>	
Mushroom toppings	\$1.00
Olives	\$2.00
<hr/>	
Extra cheese	<button>Discard</button>
\$2.00	
<hr/>	
<a href="#">← Pizza toppings</a>	<a href="#">Order →</a>

Discarding the extra cheese options removes this decision immediately.

Or, when done doing some work, applying alterations to a resource (a company detail page in this case), one can be prompted with the following:

Continue with all changes?

You made 16 changes to the company details. Continue to save all changes. Discard to remove all changes.

[Cancel](#) [Discard](#) [Save and Continue](#)

Here, "discard" and "save and continue" are in fact opposing options, and both move the user's journey forward. They both do data manipulations. "cancel" in this context is just navigating the user back to the previous view, where the alterations happened and are then still to be saved or discarded.

## "You do not take accessibility into account"

Person on the internet

**You do not take accessibility into account**

Me

**Accessibility warrants an entire article by itself, but as I believe this may be the most important part of the entire debate, let's talk about it nonetheless.**

- Accessibility warrants an entire article by itself, but as I believe this is an important subject, I'll include— what I believe — the crux of it.

First we have to address the ARIA elephant in the room. Roles are not the same as semantic elements. For instance, the [w3org spec](#) says that even a `div` can have `role=button`. But do we really want to go down that kind of semantic road of headache? We prefer to use semantically correct elements. So buttons are `<button>` and links are `<a href>`. Buttons are about actions. Links are about resource navigation. w3org says links can have an accessibility `role=button` if the context requires it. One can question if the w3org spec has it right, though, but that's a different discussion (I'm for sure willing to have in the future). Let's then put this on the "`a div can have role=button`" pile of questionable things we can do with our DOM.

MDN says the following:

The `button` role is for clickable elements that trigger a response when activated by the user. Adding `role="button"` tells the screen reader the element is a button, but provides no button functionality. Use `<button>` or `<input>` with `type="button"` instead.

A link is focusable and can be triggered by the enter key. A link will redirect you to a new page or a section within the same page. In VoiceOver's rotator, it will also be collected within the "Links" window. A button is focusable too, and can be triggered by the space key. A button will trigger an action like opening or closing something. Or sending a form. In most cases JavaScript is used to do this. In VoiceOver's rotator, it will be collected within the "Form controls" window. That alone says something.

As you can read, the definitions are a bit "interpretable". But we can extract that a "triggered response" is the effect of having data that's been manipulated. The response is the result of that user activation. So, *saving* something, or *deleting*. a11y-101 talks about links that redirect you to a new page or a section within the same page. That is absolutely true, as we've seen before. Clicking a link that says "add extra cheese" can *absolutely* navigate the user from `example.com/order/pizza` to `example.com/order/pizza/extracheese` — Again, think of what Chris Coyier said: *"If there is any kind of href you could put on that link that makes sense, by all means, use an anchor (link)."*

Where a11y-101 leaves room for debate though, is where they say: *"A button will trigger an action like opening or closing something."* A lot of the confusion stems from the fact that "cancel" or "close" in a modal view is physically (visually?) seen as closing a window. But that's where we're in disagreement. **It's merely navigating from one view (state) to another.** Again, maybe on your desktop browser, the "add extra cheese" view is a modal or side panel, but on your tablet or mobile browser, it's its own page. Is it still "closing" then? Would a button all of a sudden become a link, just because responsiveness kicks in? And what about reddit's and dribbble's presentation, which shows the user an overlay when navigating from the list view (homepage) to a detail view, but a dedicated page when navigating to that exact same URL directly. Is it closing or navigating back?



Experience Design Leader | User Experience is Brand Experience

Omfg if it is an action it is a button NEVER A HYPERLINK. I will die on this A11Y semantic hill.

Agree. If it's an action, it's a button and never a link. The question is though, is it an action, or just navigation. (We think navigation!)

**It's this ambiguity, that we want to prevent.** We want to be clear and — dare I say — consistent! Navigation warrants a link, actions a button. Regardless of your presentation, viewports or responsiveness.

Person on the internet

**What does that mean for assistive technology?**

So, establishing that semantically, we want buttons for actions and links for navigation. **What does that mean for assistive technology**, such as keyboard navigation? I had a nice back and forth with MoreThanOrange. I'll put his response in full here, for full disclosure. He said:

Your conclusions do not take accessibility into account. They make it harder for people who like myself, who rely on assistive technology. Please reassess based on inclusive principles. As a keyboard only user, I have expectations of how a button versus a link will behave and how I can interact with them. A link navigates me to a new url. I can use the space bar to activate the link. Or I can get a context menu (right click is how a mouse user activates it, but I use shortcut keys) to open the target in a new tab. It is not merely a visual construct, as you are implying. There's a difference between a form that opens on the page versus one that opens on a new page. Buttons and links set that expectation. To close a form dialog on the page, I should be able to press the escape key. But for a form on a new page, I use shortcut keys to go back. Your uninformed assumptions make my UX frustrating and difficult. So please do a little research. Talk to users of assistive technology. Because

The crux in his response is this part: "*As a keyboard only user, I have expectations of how a button versus a link will behave and how I can interact with them.*" You are absolutely right! **And it's this exact difference that has you reassured what's about to happen.** Activating a button changes data. It's a button press, just like flipping a light switch. **It's a fundamentally different interaction than clicking a link.** Activating a link does not *change* any data. It brings you to a new state of the web application. It only changes your view. And that should be reflected in the URL and the link's properties. Using assistive technology, you should be aware of what's about to happen when interacting with said link. Being allowed different types of interactions (space bar hit, enter press) reinforces that. He continues: "*A link navigates me to a new url.*" Correct!

He then says: "*Or I can get a context menu (...) to open the target in a new tab.*" Again, very true. There should nothing be stopping us from opening that "extra cheese" view in a new tab or window, if we wish to. If it's something you would *want* to be doing is a different story, but technically you're right. Another example would be a link that says "login". It could "open" a login modal view, but you could also open the link inside a new tab or window, where you would see the login view as a dedicated page (like reddit or dribbble). This behavior does not make sense for buttons with actions. He continues: "*There's a difference between a form that opens on the page versus one that opens on a new page*". There shouldn't be. Just because one opens in a modal view and the other doesn't, does not make it a fundamentally different thing. Again, think of the reddit/dribbble example or responsiveness examples.

He ends with: "*So please do a little research. Talk to users of assistive technology.*" I don't take this lightly. His response clearly demonstrates how important accessibility is, and rightfully so. Many people fully rely on this to work as expected and consistently. It's exactly for that reason that we're having these talks. As said before, this subject warrants a dedicated article. I've been going back and forth with people who use assistive tech. Together with a UX colleague, we've been doing research on optimizing UI for accessibility. I'll probably

## "This article has me thinking"

Person on the internet

**This article has me thinking**

Me

**I guess that's a good thing!**

## "This article is way too much thinking and too abstract"

Person on the internet

**This article is way too much thinking and too abstract**

Me

**While I understand the desire to no wanting to overthink things that seem obvious, in reality they aren't always obvious.**

— I think this falls into the same category as the first point "It's a button, no discussion". While I understand the desire to not wanting to *overthink* things that *seem* obvious, in reality they aren't always obvious.

Furthermore, it's our task as UX developers to think subjects like these through, up to a point where we deeply understand what's going on, so that others don't have to, and can **simply rely on solid research**. So, for sure, not every sprint you want to have a full blown academic research sprouted. Probably never actually...



Jesse Basham • 3rd+

Yes, I had a good weekend. No, I'm not currently seeking new work.

1w \*\*\*

Users don't know the technical difference between a borderless button and a text link, so it really comes down to aesthetic preferences and whatever is most consistent with your existing design system or component library. Anything beyond that is just navel gazing.

Jesse's reply discards any discussion beyond calling this a "purely aesthetic" and "consistency" matter, as "just navel-gazing". I respectfully disagree. Calling this type of discourse navel-gazing and thus disqualifying anything even remotely academic we do in the process is, in my opinion, a rather stagnant mindset. How would we evolve our field of expertise if we avoid anything that we feel uncomfortable with?

Regarding aesthetics and consistency, I already laid out what's to say about that, earlier in this article.

## "I'm concerned about Fitts's law, Jakob's Law, Cognitive load, etc"

Person on the internet

**I'm concerned about Fitts's law,  
Jakob's Law, Cognitive load, etc**

Me

**If usability is your primary concern, then I'm fully aboard!**

— If usability is your primary concern, then I'm fully aboard! Because it's precisely the reason why I ask the question in the first place. Empowering the

Fitts's Law: Nothing prohibits you from making the hit box (line-height) large enough so that it's easier to click. Jacob's Law: that's *exactly* why I'm discussing this: Not only do I want to prevent the "cancel as a button" pattern from becoming the standard, I also see that the "cancel as a link" has actually always been the standard to begin with.



Anne Jackson (She/Her) • 3rd+  
Senior User Experience at PPG

1w ...

Cancel as a button for two reasons:

- 1) it's more common;
- 2) having two differently styled buttons creates more cognitive overhead for the user.

As Jakob's law: "Users spend most of their time on other sites. This means that users prefer your site to work the same way as all the other sites they already know."

Anne mentions "two differently styled buttons" creating "more cognitive overhead". Having two different "buttons" doesn't increase the cognitive load per sé. Nor does it not. It depends on what you're trying to accomplish and how you're trying to execute on it. It *does* add cognitive load if it makes the user *wonder* what something does. By having all interactions look similar (i.e. everything is a button, regardless of function), you are setting expectations that might not be true. *Then* you are adding cognitive load to the user; they will have to engage into a guessing game. Or at the very least, they will become less confident about what the consequences are of the element they are about to interact with. Or as H Locke says, if we don't care about patterns: "*we can look forward to a world where users kind of expect anything, or that they have to work each website out for themselves*". Talk about cognitive load!

H Locke "If we no longer really adhere to design patterns at all, (...) we can look forward to a world where users kind of expect anything"

of course, it's not two buttons, but a button and a link. But we already knew that.

## "It should be a user researched decision"

Person on the internet

**It should be a user researched decision**

Me

**That I actually fully agree with. In the end, nothing beats actual user research.**

— That I actually fully agree with. We can analyze and draw educated conclusions from what we know and understand about the web, user behavior and patterns all we want. But in the end, **nothing beats actual user research**. The claims I make about reinforcing patterns which would lead to an increase in usability is—in the end — still is a hypothesis. Sure, based on other research and expertise, but a hypothesis nonetheless. Therefore, my follow-up article will be about the actual user research done to either prove or debunk these claims.



JOSH EPSTERN · 1 min read · START

Digital Product Design Leader | Champion of team culture | DEI + A...

This is such a thoughtful perspective! Although I appreciate the data behind this design rationale, I can't help but think "...do users understand this mental model?" I imagine many users see buttons and links as a hierarchical/visual difference and not a difference of result. I'd be curious to know if I'm wrong, since design patterns are defined as usable by those who use them, not those who design them. Interesting read regardless!

Like · 1 | Reply · 1 Reply



Karim Maassen · You

Founder / CEO nwzer.com, Dutch Entrepreneur, Digital Com...

1s ...

A very fair point to raise. I'm currently working on two follow-up articles, one of them being actual research done on users to see whether or not the mental modal that I claim exists (and has value) is actually measurable. So far it's been an academic story, but reality can be different.

Like | Reply

Josh, that's a very fair point to raise. I make academic claims that need validation. While building upon heuristics and earlier research (secondary research) is a good form of abductive reasoning, I would still want to be able to "see for myself" what the impact is on usability when applying (or not applying) the claims from the previous article.

## So, let's test!

Which brings us the what lies beyond this essay. Question is what to test, and how. What stands for "improve of usability" when talking about "empowering the user" with "predictable patterns"?

uxmag "Predictability in UX can be defined as how much a user can successfully foresee the result of an interaction."

Measuring a user's success when traversing a journey, would be measuring the mental maps they use to accomplish their goals. If we multiply speed by accuracy (intention), we get efficiency (we're measuring Fitts' s Law). These are measurable. But we can also measure (ask) how confident a user is when asked to

being exposed to a certain pattern, or by mixing them up. Can they foresee the results of an interaction? These are questions that we've been tackling ever since the original article. Keep in touch to be notified of the follow-ups!

## Keep in touch

Consider [subscribing for free](#) to get email alerts when I post new articles, or [join Medium for full access](#) to my article archive, plus everything else on Medium. Follow me on [Twitter](#) to see the on-going discussion!

→ <https://medium.com/@karimmaassen/membership>

#ux

#ui

#ui-design

#accessibility

#design