

User-Centered Dashboard Development

A Practical Guide for Data Analysts



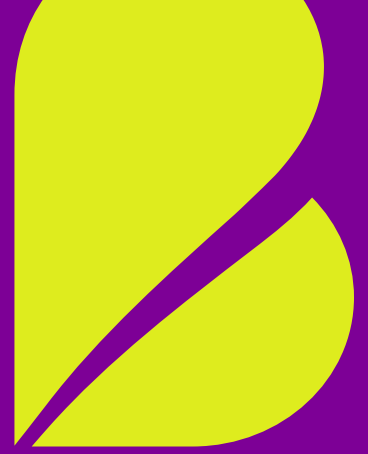


Our Mission

Bixal’s mission is to positively impact people through human-centered and digital solutions that drive equity and global social good.

We define human-centered solutions as prioritizing user needs through empathetic design that creates functional, user-friendly products and services that enhance the overall experience of the individuals they serve. This approach embraces a deeply nuanced exploration and understanding of real people—their expectations, their emotions, and the challenges they face.

Data analysts are responsible for delivering dashboards built using human-centered design, but they sometimes struggle to understand the needs of their targeted users. Use this guidance to help you gather information and transform it into a design that delights and works as intended.



Contents

1 Introduction

3 Discover

Simple User Personas

Need-to-Know Items

5 Data Discovery

7 Curate the Feature List and Backlog

9 Learning While Building

Location Matters

Cognitive Load

10 Eye Movement Patterns

12 Designing for Accessibility

13 Grouping Assets to Users' Needs

14 Visual Hierarchy

18 User Acceptance Testing

19 Deliver

Proactively Solicit Feedback

20 Measure Usage

Manage the Backlog

21 Conclusion

22 Meet the Author

Introduction

Did you know that many dashboards created and delivered by data analysts often go unused? According to a [2019 Deloitte study \(PDF\)](#), 67% of senior managers say they aren't comfortable accessing or using data from the tools at their disposal. Even at companies with strong data-driven cultures, close to 40% of respondents expressed some level of discomfort. This statistic should prompt serious reflection and redress—and soon.

An April 2023 post from [Statista](#) reports that 90% of the data stored in servers around the world were collected in the last two years. We are sitting on massive amounts of data in various states of usefulness. But the reality is that most professionals are still learning to use data for decision-making. To the data analyst, these zettabytes of new information are a playground of insights. However, our business users possess varying levels of data literacy and have diverse needs.

The number one pitfall data analysts often fall into is assuming that dashboard users are just like them. This basic (and incorrect) assumption alone leads to massive data product investment loss. Human-centered design (HCD) is crucial to every dashboard project to ensure we deliver solutions that get used. It's not the curiosity of the analyst that ought to be at the center of design, but the real pain points of the users.

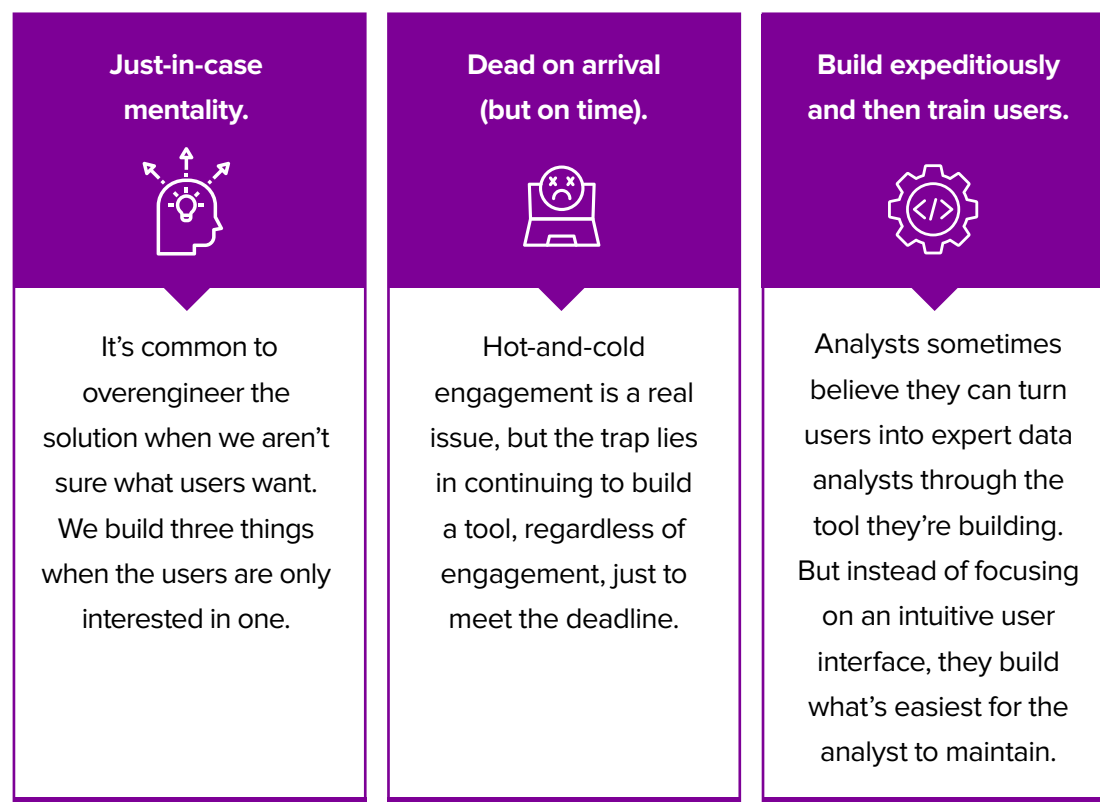
Employing an iterative, human-centered approach to product development helps to ensure that our understanding of user needs and behaviors continuously deepens and informs ongoing improvement. It is best practice to begin and end with a focus on positive customer experience. This white paper outlines the basics that any data analyst can follow on their own to achieve a functional dashboard design for their users.

When designing a dashboard for usability, never assume that a user's preferences are the same as ours as analysts. We must also keep in mind that our users are comparatively untrained, busy, and overwhelmed as decision-makers. More than anything else, our goal is to help them find what they need as quickly as possible, with minimal effort and zero confusion. We do this by fully comprehending user goals, maintaining strong and consistent feedback mechanisms, and keeping the tool simple, predictable, and clear.

No one starts a dashboard project with the intention of creating something useless. However, we often end up there inadvertently, despite our best intentions. The purpose of this guide is to explain great human-centered processes and to note common traps along the way.

No one starts a dashboard project with the intention of creating something useless.

Before getting into more detail, let's explore the three basic traps that data analysts often fall into:



Now, let's explore what constitutes a great dashboard-building process. The following sections are divided into the HCD best practices of discover, learning while building, and deliver.

Discover

Simple User Personas

The first step to building a great dashboard is identifying the top one to three types of users and the information they need to glean. User personas in this context can be very straightforward, focusing on the person's characteristics, the pain points the dashboard should solve for them, and the person's expectations of the dashboard. At this stage of development, ignore the users who are less likely to use the dashboard; these outliers can be addressed later in the development cycle through the backlog. Our initial goal is to capture 50–80% of the expected usage.

It's not the curiosity of the analyst that ought to be at the center of the design, but the real pain points of the user.

Once your core users are identified, talk to them! Ask them open-ended questions in an interview format to determine what is at the top of their mind. When conducting user interviews, focus less on the phrasing of questions and more on what you need to know walking away. To that end, create a "need-to-know" list. Consider rephrasing questions three times until the interviewee understands your intent.

Need-to-Know Items

- ▶ Current decision-making and reporting processes.
- ▶ Pain points with decision-making and reporting processes.
- ▶ The rate at which users need to retrieve data and information.
- ▶ The level of data accuracy and/or precision needed for decision-making.
- ▶ Whether decisions are currently being made without data.
- ▶ Constraints on data availability or quality holding users back from using data.
- ▶ Users' data literacy and comfort level with new technology.
- ▶ Factors that would motivate users to use data if they aren't used to doing so.

The Trap: Letting the user tell you how to engineer the dashboard and never finding out what their workflows, processes, and pain points are. We walk away thinking it's okay that we don't understand any of the answers to the above questions; they just gave the specifications. Even better! Now we just need to do what they said. Which leads to . . .

Poor Results: We are no longer human-centered collaborators, and we completely lack the context to make decisions along the way or to elevate recommendations. Without context, we misinterpret specifications and/or make poor downstream decisions. We also now feel compelled to go back to the user with every single decision, and this ends up disrupting the timeline.

The resulting user persona might look like this:

Dashboard Topic: US Business Health

Primary user: Policy advisor for a member of Congress

I would like to understand trends in the economy. I'd like to understand how different events such as natural disasters, policy shifts, and other economic shifts have impacted business health in the past. I would also like to forecast future economic health based on what we know of historical trends. I understand basic statistics and generally know how prediction models work, but I want a tool that will do the heavy lifting for me and provide regular updates. I'm interested in seeing the data nationally and by state.

To be successful in my job, I need to have a systems-thinking approach, understanding the pros and cons of different actions as they influence each other and the whole. This means I don't want isolated facts, but a view that will allow me to see potential interactions of business revenues, wages, and workforce composition. I also want to quickly see historical and projected future patterns of the same.


I'm most interested in getting the information right. For the facts in my tool, I want to know the level of reliability and statistical inferences where known. For example, I might be interested not only in how two groups differ, but also if those differences are statistically significant. My work is measured by how reliable my advice is to my boss, and if the policies we write end up having targeted benefits to our constituents.


I'm currently making my best estimations on what could happen in various policy scenarios based on an abundance of research articles and various examples from other contexts. Since no two situations or contexts are ever the same, you could say I engage in very well-informed guesswork. I believe that a tool backed by data science will serve to validate and refine effective policy solutions. I will still spend a great deal of time in research, but a tool will make the application of the research to my context more reliable, creating more effective policies.


Data Discovery

Now that you understand your users and are developing an idea of what to build, it's time to fully explore the available data. At this stage, you should pull in all the available datasets applicable to your use case and evaluate them for accuracy, bias, completeness, consistency, cadence, and timeliness. Document every useful column and variable in every dataset you leverage. This important step in documentation will drive future decisions and make your inputs transparent for future users of the tool.


Data discovery checklist


-  **Accuracy** For each variable, document the precision. Investigate the published accuracy of each data point and any reported margin of error.


-  **Bias** Identify how the data are sourced and consider whether the collection method might introduce bias or impact quality. For instance, data can introduce bias if it comes from a subgroup of the population. One example could be voter registration rolls, which do not represent the entire adult population. Your data may have been collected online and not include people with limited Internet access and/or those from underserved areas and populations. Bias can also impact your findings if data were collected using forms that presumed a certain level of literacy or proficiency in English. Biased data are still useful but may need to be augmented if you are attempting to be representative of a community.

-  **Completeness** Determine what percent of each variable is filled. Check to see that you can differentiate 0 values from missing values. Quantify invalid entries that will introduce errors and determine what percent of all entries are assumed to be valid information.

Data discovery checklist (continued)

-  **Consistency** Explore whether any data types or format issues need to be resolved. For example, if the data involve names, some might be listed last name-first name, while others might be listed first name-last name. This information is still complete, but you can make note that formats will need to be resolved later. Be sure to determine if the data points are consistent with other accepted sources of truth. This requires spot-checking a random sample against other authoritative datasets. Note the types, volume, and scale of discrepancies observed.

-  **Cadence** Find out how often datasets are updated. If the data involve periodically reported data, determine if the cadence is daily, weekly, monthly, annually, or another frequency.

-  **Timeliness** Explore whether there is a lag in the data being published. For example, a dataset might have a daily cadence, but the lag is six months. Depending on your use case, you may be fine to use data six months old. For other purposes, you may require near-real-time information.

Curate the Feature List and Backlog

Now think about which features to include in your dashboard based on user needs and what is possible with the available data. Follow these steps:

- 1.** Open a spreadsheet and make a list in column A of potential features. In this context, a feature is usually a key performance indicator (KPI), chart, graph, etc. It could also be a prediction or forecast. It could even be a download button. A feature is essentially a helpful thing you want to put on the dashboard. If you have two users or use cases for which the same variable is optimally presented differently, this is two features. Later, we discuss that it's not bad or wrong to repeat information on a dashboard. Anything that helps our users is good.
- 2.** In column B, rank the feature list according to how helpful each one will be to your users. (Be sure you are empathizing with your users and not thinking about what you would find helpful or interesting.) The most helpful feature is a 1. The least helpful feature to the user personas is the highest number.
- 3.** In column C, rank the list according to what is easiest to create. The easiest thing to do is a 1. The hardest thing to do is the highest number.
- 4.** In column D, add $B + C$.
- 5.** Sort using column D, from lowest to highest.
- 6.** Determine the cutoff point of your feature list using this cost-benefit approach. Often, in the first release, we will only do 25% of what we imagine would be nice to have.

Now that you have decided who your users are and what is feasible for you to create for them, you can start designing the dashboard layout.

Features	Helpfulness Score	Ease of Creation Score	Totals
Download Button	1	1	2
KPI	1	2	3
Chart	3	1	4
Prediction	2	4	6

In the above example, KPI was deemed the most helpful dashboard feature as it would summarize key results quickly. The predictions feature ranked second in importance, but it would require complex behind-the-scenes development.

Learning While Building

Learning while building in the context of HCD refers to the iterative and hands-on approach of gaining insights, understanding user needs, and refining solutions through the act of creating and testing prototypes.

We recommend taking an agile approach to building the dashboard and delivering value as quickly as possible. Do not make users wait for weeks or months to start getting value from you as a data analyst. If your users are desperate for a few facts, do the math! Make sure they have their answers and then, afterward, build them a tool that delivers answers on demand. It is critical to deliver value quickly, build trust and rapport, and then build iteratively with frequent feedback.

Once you've established a clear feature list for the initial build (set the backlog aside for now), present it to the customer. Accompany it with a design that visually illustrates how the features will function. Excel can be a great resource for creating quick charts from subsets of data. You can arrange these charts in a worksheet or in a PowerPoint presentation to begin visualizing the dashboard's appearance and functionality.

Location Matters

When designing a dashboard for usability, creativity is much less important than ease of use. Above all, we want users to find what they need as quickly as possible, with minimal effort and zero confusion. Aligning your design with the typical patterns of our human eye movements is a pivotal component of usability. It's not about filling every space; it's about the strategic placement of your assets within the space.

Cognitive Load

Cognitive load refers to the amount of working memory a person employs to perform a task. Think of it as the random-access memory (RAM) in a computer—we humans essentially have RAM in our brains. It is vital to design dashboards that minimize our users' additive cognitive load. Much like computers, our brain's capacity varies based on our innate abilities (hardware) and our current cognitive load, which can be influenced by other ongoing tasks and stressors in our lives (concurrent processes). Not only do we show up with a certain intrinsic capacity to read dashboards, but that capacity also depends on what else we're thinking about when we attempt to use a dashboard. Refer to your user interviews and empathize with their data literacy levels as well as their stressors and pain points. Our basic tools should work for us even on our hardest days.

By designing in a predictable pattern, the user is put at ease. If you want the user to consider two data points at the same time, put them next to each other. Don't make users hunt for what they need or force them to toggle back and forth excessively. Lastly, in the pursuit of designing a dashboard to eye patterns and convenient groupings, don't worry so much about the white space. White space will naturally happen—and it's a good thing. We'll dive a bit deeper into each one of these crucial points in the following sections.

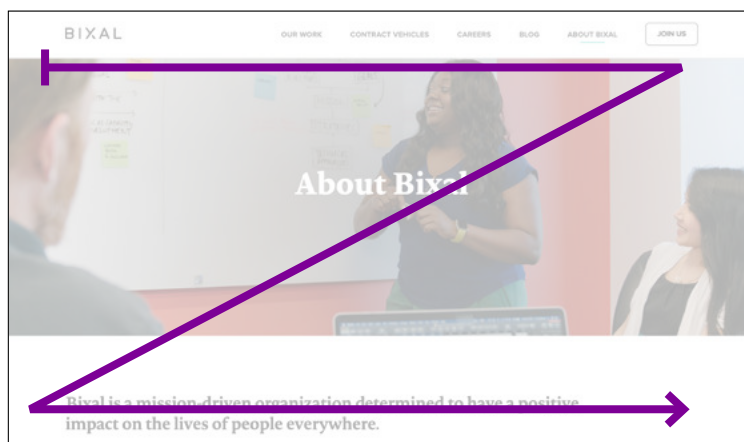
Keep it simple, predictable, and clear.

Eye Movement Patterns

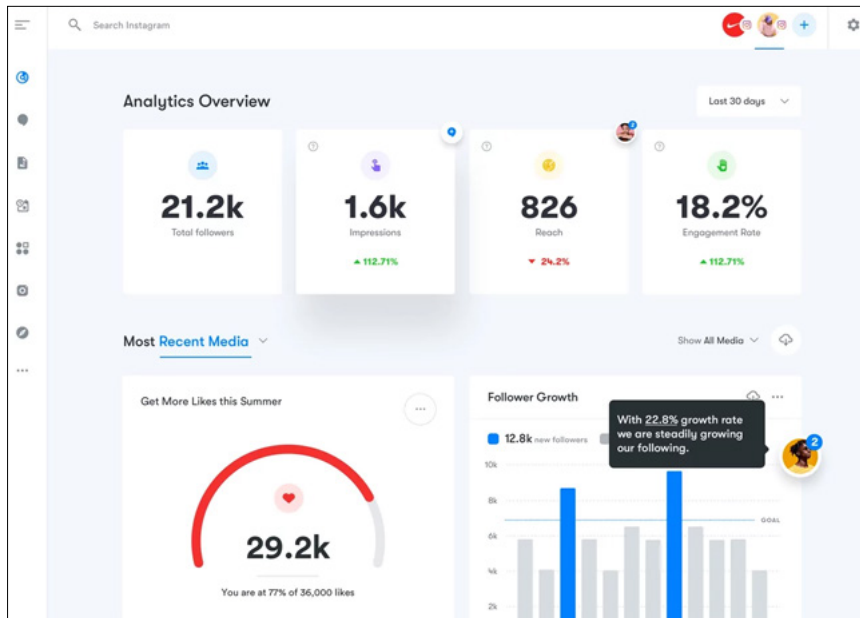
Z-pattern

The Z-pattern is commonly used for web pages with small amounts of text or content. It works best when you need to make a specific point or convey a conclusion. The following example is a landing page from the Bixal website that is designed to help the user follow this eye pattern. Starting at the top left, users will see the logo, then the navigation tabs, then cross back over the image and see that this page contains case studies, then finally to the bottom where they read the details of the text below the image.

Generally, what you put at the bottom of a Z-pattern design are the details an interested reader (someone who gets past the headline and is thoroughly engaged) will want to know. When using this pattern for data visualization, place the important headlines at the top, and any "fine print," such as data sources, date stamps, or minor explanations, at the bottom right of the design.



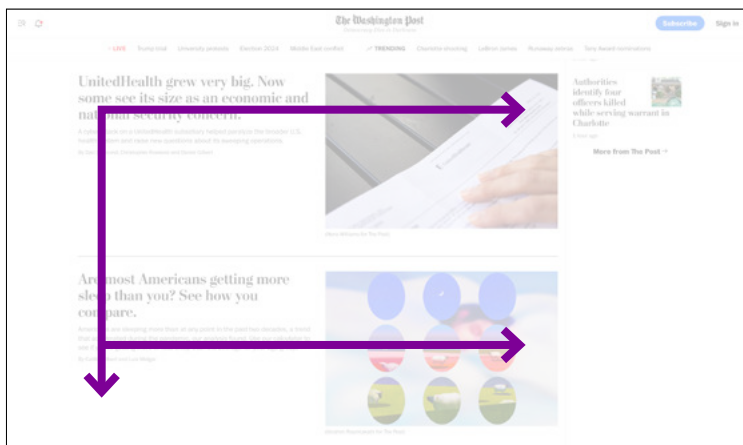
In dashboarding, the Z-pattern can work when your focus is on KPIs. You can put these important stats across the top for easy scanning, like this example dashboard:



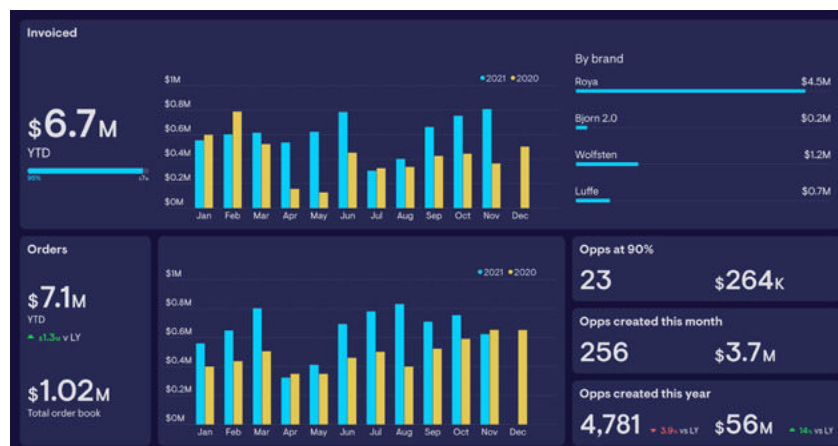
Source: [JustinMind](#)

F-pattern

The F-pattern is commonly used for web pages with large amounts of content. With this pattern, the user scans down the left side of the page for content to engage with. When they see something they want to know more about, their eyes move to the right to capture the details. In the following example, the website has headlines on the left side of the page to catch the user’s eye as they scroll down.



This eye movement pattern can be leveraged for complex dashboards or data stories. When we leverage the F-pattern, topics evolve as the user scrolls down the screen. In the following example, the KPIs are to the left and the detailed stats are to the right of the summary KPI. Here the user can find a KPI of interest and then follow the theme to the right to see more detailed data on that topic.



Source: [Geckoboard](#)

Designing for Accessibility

It's important to ensure your dashboard design is accessible to people who are blind or have low vision. Because many of these users employ the use of screen readers, make sure the flow of content is easy for screen readers to follow using heading levels and other accessibility best practices. Screen readers work exactly how Google does; they start at the top and work their way down in order of the document object model.

If your dashboard is in an iframe, accessibility features will not be readable by assistive technology. In this case, you can put a link outside the iframe to either download the data or to navigate directly to the original asset, in an accessible form.

There are more accessibility principles that analysts can apply than we have the space to cover here. For tips on how to make data more accessible, explore WCAG's [data visualization guidelines](#). To learn more about designing for accessibility, see [accessibility.digital.gov](#).

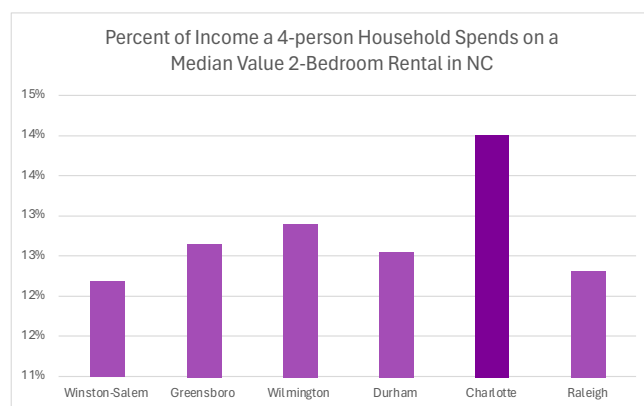
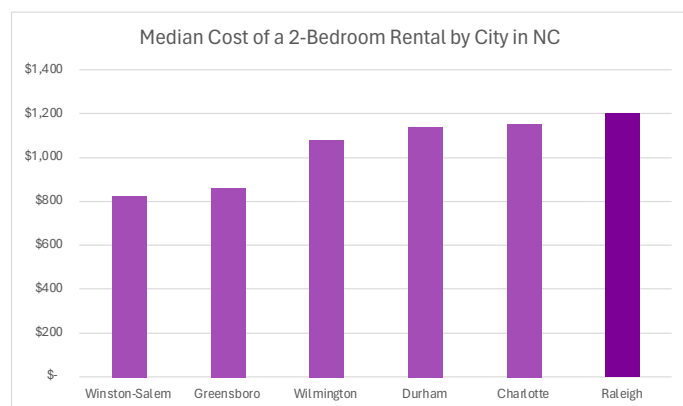
Provide alternate ways for users to access your data. If your dashboard design doesn't comply with WCAG 2.1 standards, give users the option to download raw data in a spreadsheet or other format. This way they can customize the data to suit their needs.

Grouping Assets to Users' Needs

How we place items on the dashboard in groupings comes from both understanding the data and understanding the user personas. Who are my users? What are they here to find? What order and which groupings of data work for each user persona?

People often intuit that things close together must be similar. Proper location is not only a usability issue, but it also helps users correctly interpret the data. Sometimes we need a user to see two data points next to each other for them to walk away with information that will lead to the correct decision or action. It's more than just good design; it's key to the tool's purpose.

As an example, let's say you are assessing the housing prices across the country to determine what city has the most severe affordability problem. What if we only showed the median price of a two-bedroom rental across the entire country? We would infer that the most expensive city is the least affordable. But is that true? Maybe, and maybe not. We do not know for sure without placing a median salary for a four-person household with the rental price for the two-bedroom housing unit. Expensive cities also tend to support higher salaries as the following charts demonstrate:



Sources: huduser.gov, National Association of Realtors

In summary, always design to the user's persona and purpose. Be consistent, but also thoughtfully break consistency when it is vital to the user's understanding.

The Trap: Never repeating information and considering redundancy to be a waste of time and effort. Making the user click extra times to find what they need instead of repeating a feature in two places. Which leads to . . .

Poor Results: Our user is either short on time or forgets all the places to find their information. They miss vital information because the efficient design doesn't fit their flow.

Visual Hierarchy

In the previous section we talked about how the eyes move in predictable patterns across a dashboard. Visual hierarchy is how our eyes know where to pause as they scan. Users should be able to quickly land on the content as they are looking. If your eye is jumping around and doesn't know where to land on the screen, it's a sign you either didn't follow a natural pattern or you need to improve the visual hierarchy.

Before you begin building your dashboard, you should know the most important "stops" on the users' journey through the tool. For each of your users, you have listed what they need to get from the tool. Help them find what they need by making headlines stand out.

Chart font type, size, and color

Headings or titles should be large and bold. Headings should also be semantically clear. The largest, boldest font should be the H1, or the first heading you use. To improve accessibility and to ensure consistency, determine if your dashboarding tool allows you to designate semantic headings that can be used by assistive technology.

Subtitles are a little smaller, should use the same font, and should go under the headings to clarify what's being shown. Text that elaborates on the details of the chart or graph are the smallest. Imagine your user is nearsighted and viewing from a small screen when you choose the size of the body font. In other words, be kind and don't make it too small. **Do not use more than two fonts on one dashboard.**

Regarding color, the headings and subheadings should be chosen from the darkest color of the branding palette you have for your dashboard. The explanatory text should be either black or dark gray, depending on the palette as well. Which specific fonts you use are variable by the tool you are using to build the dashboard, but below is an example of mixing Arial with Verdana:

Heading 1 (h1), Arial 18 pt bold

Heading 2 (h2), Arial 14 pt

Explanatory Text (p), Verdana 11 pt

Ensure you have proper color contrast with fonts against their backgrounds. If providing the data for a federal client or group, the color contrast should have a contrast ratio of at least 4:5:1 against the background color. You can measure the contrast with a tool such as the [WebAIM contrast checker](#).

By following these guidelines, you are using size and color as tools for visual hierarchy. The headings and subheadings stand out because they are a dark non-black color. They help the user's eyes stop on the feature of most interest to them. The explanatory text is more subdued and appears as something that can be read later if they're looking for more information about the data being presented.

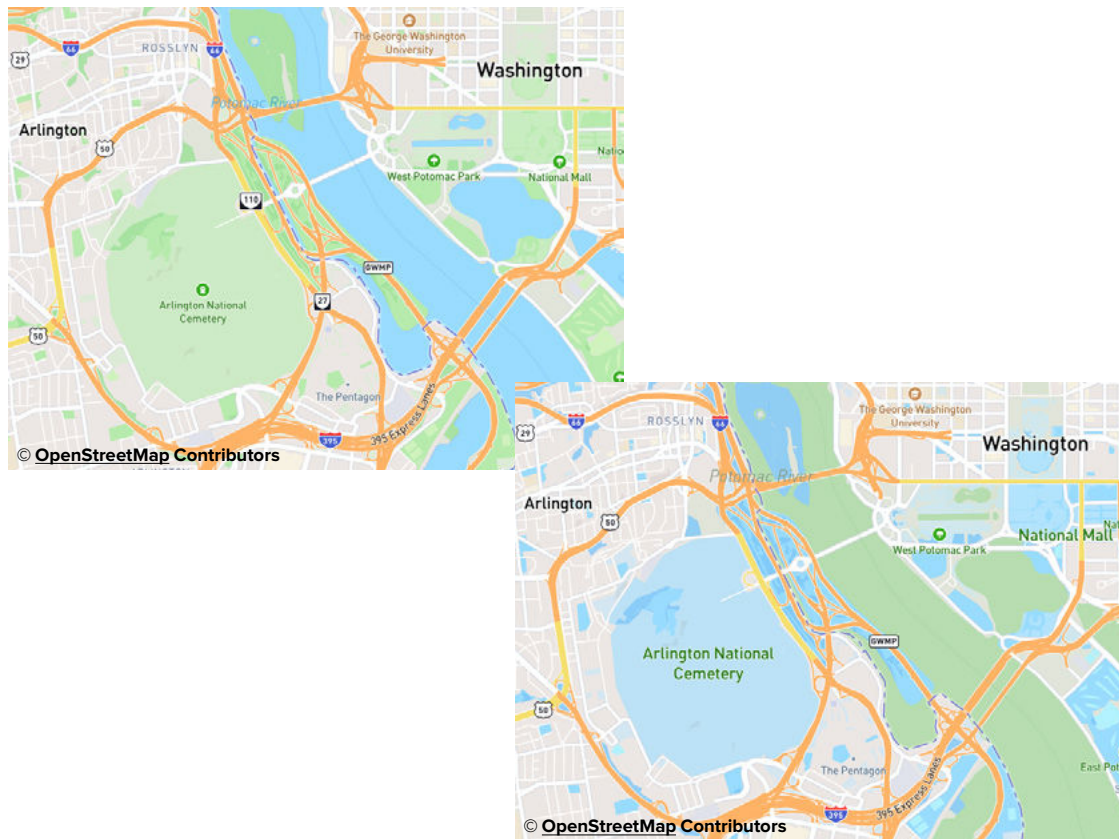
This advice comes from best practice and research that speaks to most people. When considering accessibility, the needs of individual users could vary greatly. If someone is low vision and using a screen reader, the heading levels structure the content for the reader. Use levels (H1, H2, etc.) whenever possible, understanding the screen reader will work its way from top to bottom. [According to Siteimprove](#), the most common accessible fonts are **Tahoma**, **Calibri**, **Helvetica**, **Arial**, **Verdana**, and **Times New Roman**. With that said, many users with accessibility constraints leverage tools that can change the appearance of text or read it out loud. Make sure your headings are descriptive and provide helpful information. Use accessible fonts when you can, but most of all, use the accessibility options of your dashboard software to ensure that whatever you present can be read by assistive tools.

Color holds meaning

Analysts are often given a brand guide to follow when making a dashboard. Use the branding you are given, but keep in mind that when it comes to the presentation of data we associate color with topics, emotions, and a sense of action.

Not only do our brains have a finite cognitive load, but we also wear cognitive glasses. Our cognitive glasses are our expectations of how the world works. Asking the user to change their cognitive glasses significantly increases their cognitive load—assuming they are even capable of wearing the new glasses you just gave them.

Think of a web mapping tool—a common type of data visualization. What would happen if you made the lakes green and the parks blue? Your users would be very confused. They would have to constantly tell their brains green means water and the parks are not lakes.



Source: [OpenStreetMap](https://www.openstreetmap.org/)

We come to our screens with certain expectations about color as a function of the cognitive glasses we wear. A great dashboard leverages these expectations for readability; it does not force the user to mentally adjust. To that end, here are some colors you should use deliberately in your data and colors to avoid in your overall color theme.

Shades of Red	Shades of Green	Pure or Neon
<p>Red is often used to highlight a data point that is a concern for action. When we see red, we associate it with a red flag, stop sign, red light, etc. Keep the message of this color by strategically using it to send a message to stop and take notice.</p>	<p>Opposite to red, we associate green with data points that are amazing. All systems go, we have the green light, etc. Use green to send the message that things are going well, and the current state is good.</p>	<p>These colors make us anxious and are hard on the eyes, so we should avoid using them altogether. A rule of thumb is to avoid 100% saturation with 100% brightness.</p>

To fully investigate colors, one good option is [Color Brewer](#). It has toggles for different types of palettes as well as colorblind-safe options.

<p>The Trap: White space is bad, so we need to fill all the spaces with something. Which leads to . . .</p>	<p>Poor Results: The user has trouble focusing on one thing at a time. White space is our friend that keeps the dashboard looking crisp and our eyes pointed in the right direction along the Z- or F-pattern you have chosen. Pay attention to the lines and natural eye movement and worry less about white space.</p>
--	---

User Acceptance Testing

Optimal practice before delivery is to conduct two rounds of testing—moderated and then unmoderated—to flesh out any critical issues regarding the data, programming, and/or function of the tool.

In moderated testing, we give the user a task to complete with the tool. The moderator sits with the user and has them talk out loud while they attempt to accomplish the task. In the moment, the user can express their thoughts around their interactions with the tool. For example, they might note anything unclear, ask about any overly technical terms, or pinpoint onerous navigation issues in the moment. The moderator takes notes and notices patterns across users that inform critical adjustments.

Unmoderated testing is an independent exercise where users are given specific tasks and asked feedback questions. Both passes and fails should be noted with unmoderated testers as well as any difference in their user environment—such as device type and browser type. The analyst should pay attention to all the fails noted and either update the tool or add to the backlog. It's very important to retest after each update to ensure your fix has not introduced any new bugs. Unmoderated testing should be repeated until no flaws are found, and all tasks are passes. When gathering users to include in testing, strive to include a diversity of data literacy as well as any types of disabilities that could be included in your early adopter cohort. Having a diverse user testing group is key to ensuring your data is as accessible as possible.

The Trap: Trusting yourself and others close to the data tool to find bugs. Which leads to . . .

Poor Results: We send bugs to production because we didn't use fresh eyes. We continue to not see existing bugs and errors, and we also miss unclear navigation or unclear terminology because we know how the tool works.

Deliver

Proactively Solicit Feedback

Customer support is meeting the basic needs of customers in the short term. In customer support, we quickly resolve issues and determine ways to avoid them in the future. Customer success is a long-term strategy for maximizing the use of a product and maintaining a strong customer base. Focusing on customer success pushes us to work through the backlog, which improves the tool's usefulness in every aspect of the customer's journey. Customer success ensures that products stay relevant and evolve with users' needs. **Both are important for fostering relationships with users based on trust.**

Never deliver the product or new feature, walk away, and wait. Every new tool and feature needs to be thoroughly socialized. Even after user acceptance testing and refinement, it takes time for new tools (or old tools with new features) to catch on. Most people won't tell you things unless you ask. Remember, we must assume that users are untrained, busy, and overwhelmed. They may have forgotten what you said or forgotten that the tool exists. Most of us need to be told new things multiple times. People also need time to sit with a new tool to figure out how to incorporate it into their everyday activities. They won't know all the ways in which this new tool will be useful at first.

The Trap: We don't want to bother people by soliciting feedback and/or our project manager says it isn't needed. So, we don't. Which leads to . . .

Poor Results: No one uses the product after it's delivered, and we don't know why.

Measure Usage

The best measure of success for any tool is how many people use it. Set up ways to monitor usage at launch. Measure the number of users from various user persona groups, what they access, how often they access it, and how long they interact with the tool. Note changes as you socialize. Keep doing what works in terms of introducing the tool to the organization.

The Trap: There isn't enough room in the budget to measure usage, so we can't do that. Which leads to . . .

Poor Results: When attempting to justify the budget moving forward, we have no data to support the value of the tool, and the budget is cut.

There is always enough time and budget to gather data that will lead to a return on investment. Manage the budget from the start to leave space for usage monitoring and continuous improvements. Make it a non-negotiable policy for the definition of success.

Manage the Backlog

Give users a chance to use the tool, then reach out to them after a few weeks and find out if the tool is meeting their needs, and what they love about it. Also ask if there's anything about the tool that isn't working for them and consider making changes. Find out which features no one uses and consider removing them. Ask them about users who might interact with the tool only occasionally and how it could be of more use to them.

Each time you update the data or update a feature, notify people about the change. Note if they are opening your emails as well as whether they are clicking to see the update.

Check in every few months to see if the user personas have changed, if new users are emerging, and if anything needs updating. Whenever possible, conduct interviews to determine how user needs have evolved. Another way to navigate the changing landscape is to stay abreast of any changes to the way your users make data-driven decisions. If user needs evolve due to changing requirements, suggest new features proactively.

Conclusion

Dashboards, although widely used, often fall short of their full potential. Creating a dashboard goes beyond visualizing data in an appealing manner; it necessitates careful consideration of the specific data needs of a particular user group and the most effective way to organize and present that data for easy accessibility. Dashboards should remain focused on the intended audience, providing answers to their most common questions while indicating areas where further exploration of the data is required. For true effectiveness, dashboards should offer contextual information alongside the presented data and allow for flexibility for users who may need accommodation.

Treating dashboards as a reference library for team members who seek answers empowers them with valuable resources. Dashboards should not be isolated repositories of data; they should provide a targeted perspective, filtered according to everyone's role, within the broader analytics toolset. Dashboards complement data; they do not replace it. A dashboard fulfills its purpose when it offers solutions to common queries, highlights issues or opportunities, or prompts users to delve deeper into the analytics platform for additional analysis.

Meet the Author



Anna Tapp is the director of growth analytics for Bixal. She has spent the last 20 years using data to understand the intersection of people and place. She has a bachelor's degree in sociology, a master's degree in geography, and decades of experience helping organizations of all sizes make data-driven decisions. She is a passionate evangelist for using data for good, building responsible algorithms, and applying data advancements to the world's most critical issues. [Find Anna Tapp on LinkedIn.](#)

About Bixal

Based in Fairfax, Virginia, Bixal is a mission-driven consulting company working alongside governments and organizations to help them deliver better services and experiences to the communities they serve. Using evidence-based knowledge and technology, we empower clients to deliver on their missions more effectively by fostering a culture of learning and continuous improvement.