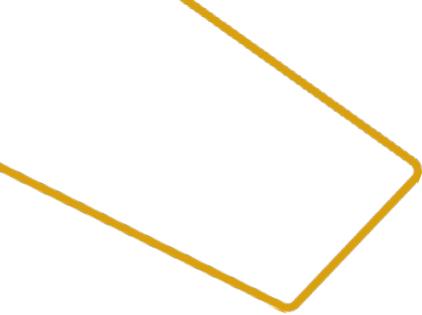




Data-driven User Guidance & Guidance-driven User Analysis

A Pendo Manifesto



In their seminal study of computer users, researchers at the University of Texas El Paso discovered that about 5% of software usability problems are successfully resolved by consulting online or offline product documentation.¹ Or to put it more simply, “users don’t read the manual.” This is not a new problem, but in the world of software as a service (SaaS) and recurring revenue where today’s software providers compete, it is a much more acute one. Successful products today are ones that deliver value to users quickly and keep them engaged over time. Customers that struggle with usability issues, or can’t become proficient with application functions, certainly don’t realize value quickly and are likely to churn.

One of the ways to address the challenge is to make as much of the user guidance part of the user experience as possible. Users are much more likely to consume content that is directly part of the application rather than moving outside to consult documentation or search the Web for an answer. Help delivered in the form of a tooltip or an in-application walkthrough can have a 30x higher engagement rate. The remedy, then to users who don’t read the manual, is to embed the manual directly within the user experience, and a number of development libraries and tools have emerged to help product teams facilitate this.

The problem is that while in-app messages are a valuable way to train and provide users help they are not perfect. They are, by design, intrusive, and can negatively impact the customer experience – especially if the help is not particularly timely or relevant. When presented with a walkthrough, 70% of users will dismiss the experience before advancing a single step.² A worst-case scenario is to make a significant investment in moving help content in the application, only to frustrate users and negatively impact the customer experience.

A better approach is to make application help truly “in-context” not just “in-application.” In this approach, the complete context of the user – especially the goals they are trying to achieve – is used to tailor the help that is delivered. Highly targeted content is much more relevant to the user, and helps to address usability problems without over cluttering the user experience. At the same time, the total user context can also be used to evaluate the effectiveness of guidance, not just in terms of engagement with help content, but in terms of the actual behavior of users in the application. This is the approach that Pendo takes with in-application guidance. By pairing in-application guidance with robust product analytics, Pendo enables users to deliver highly-targeted content to users, and then measure in detail how that guidance affects their behavior.

¹ *Why don't people read the manual?* University of Texas at El Paso DigitalCommons, 2006

² Average abandonment rate from all user walkthroughs deployed in Pendo, 2016

The power of user context

What is user context? Simply, it's everything that we know about a user in a software product at a specific moment in time. Context is tremendously valuable as it tells us a great deal about who the user is, and, what it is they are trying to achieve at that specific point in time. The more context that we have, the more we can do to improve user experience and outcomes in the application. We can think of context across four dimensions: user demographics, historical behavior, current task, and sentiment or satisfaction.

USER DEMOGRAPHICS

Context that falls in the “demographics” bucket includes basic information about the user, who they are, their locale, etc. as well as any additional customer-level detail that may be available. Assuming that we're looking at a logged-in user in a commercial application, that detail can be extensive. Demographic context can include:

- Paying or trial customer
- Plan level
- Annual recurring revenue (ARR) for that customer
- Role in the application
- Functional job role

HISTORICAL BEHAVIOR

“Historical” context includes all of the user's past behavior in the application. Here we're basically looking at what a particular user has done in the application from the perspective of both time and activity. Historical context can include:

- First login
- Average time in the application
- Login frequency
- Features used
- Task completion rate
- Number of support tickets filled

CURRENT TASK

This component of user context looks at the task that the user is engaged in a specific moment. This can include the specific page in the application that is being used, a specific feature, or even a step in a defined task.

SENTIMENT

This context is an understanding of the user's overall satisfaction with the application. This can be captured via feedback, Net Promoter Scores (NPS), or even a health score imported from a customer success solution from Gainsight.

When brought together all of this context provides a deep understanding of the user's goals, their experience and/or skill, and their overall happiness with the product. All of this context can help to determine not only whether to offer them additional guidance, but what content to emphasize within the guide.

Targeting in-application guides based on user context

All of the captured user context provides an opportunity to deeply customize guidance that is delivered to the user. But how should we think through the targeting approach? One way is to start with the following questions. Let's think through what we know about the user:

- Have they been using the application a long time or are they new?
- Based on functional role, are they more likely to be technically savvy?
- What features or functions do they have access to based on their user role or plan level?
- Have they used this particular feature before?
- Are they working on a task where users generally struggle in the application?

The answers to these questions can help to form the ways a guide could be targeted. Someone who has used a feature multiple times (successfully that is), probably doesn't need to be prompted for any guidance at all. Someone who is technically savvy might need less guidance, or perhaps explanations of additional, more advanced options. If they do not have access to certain options, then those options could be left out of a shorter walkthrough. A user with low overall satisfaction or health score might need additional help across the board. There are a wide range of ways that user context could influence how guidance is presented, and it may take some experimentation to find the right balance. It's important to produce tailored content, but the extent of the customization needs to be balanced against the time invested in customization and the ongoing maintenance of the content.

Pendo simplifies the process of targeting in-application guides through the use of user segments. Because Pendo captures all of the user actions in the application once it is installed, most of the user context is already available. Additional pieces of demographic information can be passed to Pendo directly or via tools like Salesforce and Gainsight. Segments can be defined using any combination of demographic, behavioral, or satisfaction attributes, and then each guide that's deployed in the application can be targeted to a segment. When a user is in the application, Pendo will determine if they are an appropriate target for the guide, and only show it to those who meet the criteria. All of the display logic and segmentation rules are handled by Pendo, allowing companies to focus on the help content itself.

Why invest energy in targeting in-application guides?

The idea of delivering highly-targeted help is one that resonates with nearly every software user, but there is definitely a cost involved. It's certainly easier and faster to make all of the help content follow a one-size-fits-most approach. But is the trade-off worth it? Well, one key factor is relevance. As discussed in the introduction of this paper, on average 70% of users won't advance past the first step in a tutorial walkthrough. Users simply won't engage with content, unless it's clearly relevant and valuable to them.

Another important consideration is the length of the in-application content. If an in-application training experience shows all of the possible feature help to all segments of users that means that each individual guide or walkthrough must cover quite a bit more content. Predictably, however, completion rates degrade pretty rapidly as the number of steps in a guide increases. On average, by the time you get to the sixth step in a walkthrough only about 6.4% of the visitors who started on the first step will still be consuming the content. Considering that the ultimate objective is to help users become more proficient and realize more value from the application, presenting more content than they process in a session isn't helpful.

It is also important to consider the "clutter" factor that can arise from in-application guidance. Negative or white space in a user experience is extremely important to discoverability, and proper use of white space within the user experience can increase user comprehension by 20%.³ The more cluttered the user experience, the less likely it is that users will be able to discover and learn features on their own. Every additional guide that is added (especially those that show automatically) adds clutter. There is an important balance point when the added "help" actually detracts from a user's ability to understand and achieve proficiency in the application.

Targeting in-application guides isn't just a matter of making content more relevant to users. It also helps to ensure that the user experience isn't being degraded at all by the additional content. When guides are properly targeted, users only view content that matters to them, and only when it's needed. This ensures a high level of engagement with the content which drives the actual user behaviors that increase user satisfaction and realized value.

Analyzing outcomes

Speaking of outcomes, thus far we've discussed the importance of targeting when it comes to in-application guidance, but there's another way in which the confluence of guidance and accumulated user context can be beneficial. That has to do with understanding and measuring outcomes. Let's step back and reflect on why we would want to offer help at all. User guidance (regardless of form) is designed to help users become proficient with the key functions in an application. How do you know if a user is proficient? Proficiency, measured through user activity in the application, and overall volume of activity – as well as features used – are good indicators of user proficiency.

These measures are part of the historical “behavioral context” that we discussed earlier. Capturing these aspects of user behavior over time provides a platform to measure the effectiveness of any newly introduced guidance. With improved help, do users use the application more often? Do they use more (more features or components) of the application? Ultimately, this is the benchmark of effective help.

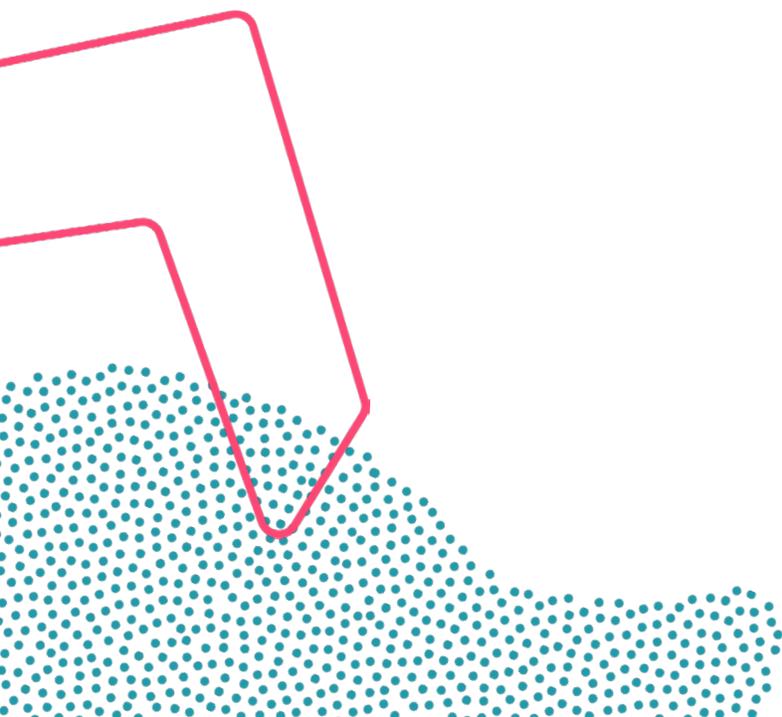
In the absence of this context, many organizations rely on actual engagement with guidance as a proxy for effectiveness. Clearly it's important to measure whether the help content is viewed, either through external help center pageviews, or clicks / views of in-application guides. However, knowing that someone has viewed help content is not the same thing as knowing that someone has learned how to complete a task as a result of viewing the content. By analyzing behavioral data in conjunction with a user's engagement with help content, companies can get a much better view of whether the guidance they are developing is having a positive impact on users or not. Given that unnecessary in-application guidance should be avoided, it's important to continually assess the effectiveness of any content delivered in the application, and remove things that are not adding value.

Pendo allows organizations to easily monitor both of these metrics: engagement with in-application guidance, and user outcomes in the same interface. Feature usage can be overlaid on guide viewership which allows for easy visibility into the impact of any particular guidance. After adding some tooltip-based help to a feature, did that feature usage go up or down? Pendo tracks that information. In the same vein, Pendo allows user segments to be defined based on whether they have viewed a particular piece of in-application guidance. This allows user behavior to be compared across cohorts who have engaged with help content vs those that haven't. This type of visibility can illuminate whether guidance is having a positive or negative overall impact on the customer experience.

Data-driven guidance elevates the customer experience

Truly “in-context” user guidance can have an incredibly powerful impact on the overall user experience. Knowing that most users will rarely consult documentation if at all, software providers who can’t deliver a better help experience are left to hope that users will be able to sort out their own usability problems. In the new area of recurring revenue, this isn’t a particularly safe bet. Moving guidance in-application and targeting it appropriately based on the user context ensures that help is relevant and available when needed, but also guards against introducing any additional usability issues through UX clutter. Capturing and measuring user context also provides a rich foundation of product data with which to benchmark the effectiveness of the content itself.

At Pendo, we understand the value and power that comes from leveraging accumulated user context to target and assess the guidance that is delivered in the application. Our platform is designed to capture as much user context as possible without any significant engineering overhead, and makes that context readily available to support product teams as they are building out help content. In the same integrated interface, Pendo allows actual product usage to be analyzed in conjunction with guide viewership. With Pendo, software companies can easily target their guidance, and measure the impact of their guidance on the user experience.





Pendo was founded when alumni from Rally, Google, Cisco and Red Hat combined their heads and hearts to build something they wanted but never had as product managers—a simple way to understand and attack what truly drives product success. With powerful analytics, in-application user feedback, and contextual guidance designed to help companies measure and elevate the customer experience within their applications, Pendo is on a mission to improve society’s experiences with software. Pendo is proudly built in North Carolina, but engineered to scale globally without losing that soul. For more information, visit: www.pendo.io.