How We Use Minimalism and a New Form of Task-Oriented Help, WalkHub, to Overcome Cognitive Load in Web Applications

Kristof van Tomme & Diana Lakatos, Pronovix

Contrary to what some software developers would want you to believe, the same principles for good documentation such as minimalism, structured content, and topic-based authoring still hold on the internet. In fact once you deal with some of the misunderstandings around the importance of documentation, there are some major trends that are making more relevant than ever before the documentation strategies technical writers developed to cope with products that are constantly changing.

DEBUNKING THE UX FABLE
To help users get stuff done, experts came up with the concept of user experience (UX) design, that incorporates visual design, information architecture, interaction design, and usability to positively impact the overall experience a person has with a particular interactive system and its provider.

UX can be the difference between a failed and a successful project. Some UX experts, however, imply, or even go as far as stating that, if the user experience is designed well enough, documentation is unnecessary.

The problem with this premise is that the familiarity of a user with an interface is always a function of a user's previous experience with similar interfaces. As a result, UX works best for people with more Internet experience. If a user has no or very little experience with online interfaces, she will still be lost.

COMMUNICATION TRENDS THAT ARE EXACERBATED ON THE INTERNET
The Internet is at the forefront of technical communication. Intense cognitive load, frequent updates, community involvement, and collaboration call for new approaches in documenting online interfaces.

COGNITIVE LOAD ON THE INTERNET
The Internet floods its users with an unprecedented stream of information that has significant implications for the attention span of users. Digital natives are impatient and need instant gratification, they have little time for deep thinking, and they don’t take the time to learn. Social media have changed communication among organizations, communities, and individuals. Twitter’s 140-character limit on updates had an impact on online communication that reaches far beyond the boundaries of the microblogging service.

The rules for online content consumption are brutal: make it quick, make it relevant, otherwise no one will care.

According to Steve Krug in his book Don’t Make Me Think (New Riders Publishing, 2000), Internet users don’t think. They use the first available solution that solves their problem. That means that it is really important to design user interfaces and the user experience so that users can accomplish their intended tasks as directly as possible. Krug emphasizes brevity and well-focused writing to support the user who is not trying to figure out how things work, but wants to simply accomplish a task. If it works—be it by simple guessing or something she has learned from previous experience—it will suffice. If it doesn’t work, she will abandon the page.

When organizations try to assist their users with manuals and video tutorials, they often increase the cognitive load. Users have a separate piece of documentation. They have to switch back and forth between a window with the documentation (text or video) and the window in which they have the software open. Users can either try to interpret and remember several steps at a time or switch back and forth endlessly.

Often these documentation artifacts are not purely task oriented, but combine background information on implementation and product concepts. When documentation gets in the way, users can’t get things done.

AGILE DEVELOPMENT AND FREQUENT DOCUMENTATION UPDATES
For documentation to work, it must be kept up-to-date. Visual elements like screenshots and videos are important to assure readers that they have found the right instructions to perform a task. They are therefore a crucial part of any documentation, but keeping them up to date is unfortunately a very time-consuming task. Even small changes in the interface can be included in many related tutorials, so organizations have to make a difficult choice: spend the time and money to update all their tutorials every time software is changed or confuse their users.

When the visuals in documentation materials are outdated because the screenshots have changed, the product design has changed, or new features have affected a certain function of the product without actually changing the core behavior, this results in a massive cognitive load for the users: They need to abstract the
HOW WE USE MINIMALISM AND A NEW FORM OF TASK-ORIENTED HELP, WALKHUB, TO OVERCOME COGNITIVE LOAD IN WEB APPLICATIONS

For consumer products, keeping documentation up to date is more straightforward. Once the product is sold, the user manual rarely needs to be updated. With software, however, updates after shipping are the default rather than the exception, and more and more products have software components. Household appliances like a smart meter or a thermostat that need to be accessed through an online user interface are only the beginning.

Agile development takes updating to the extreme. It uses iterative and incremental development methods, where requirements and solutions evolve through team collaboration. Agile development makes it possible to respond rapidly to change and ensures customer satisfaction by close daily collaboration between business people and developers and frequent delivery of working software. Some teams like Facebook update their product several times a day.

COMMUNITIES AND COLLABORATION

Many of the most successful software projects on the Internet today are developed by a community, instead of a single organization. If you use the Internet, you are using open source software daily. Most of the websites you visit run on Apache web servers, the databases behind them are handled by a MySQL server, and the website’s content is administered in Drupal or WordPress or some other open source CMS.

Open source software is usually developed in a public, collaborative manner and is often modular to make it easier to collaborate and reuse. The first versions of software are released as early as possible to find co-developers and users, whose feedback is highly valued. Code changes are merged into the shared code base as often as possible to avoid the overhead of fixing a large number of bugs at the end of the project life cycle.

Documentation for open source projects, however, is typically written by a small set of people. Individual topics might be contributed by the wider community, but the tools available to communities to create documentation often do not allow many people to work together effectively. After production, the resulting manual is treated as a document: individual paragraphs might be copied and pasted into another documentation deliverable for a project, but this is a manual process. There is no structured collaboration on topics that could be reused across a community.

So what works perfectly for code, fails miserably for documentation.

THE RISE OF STEPPED CONTEXTUAL HELP

The tools and processes used to document websites and web applications are not able to cope with the continuous barrage of changes that agile development teams create. Software developers have been experimenting with different approaches to help their users. Contextual help and more recently stepped contextual help have been gaining popularity. Contextual help is activated at a specific point in the software and specific to the state of the software, so that this type of documentation makes it easier for users to find the information they require. Stepped contextual help is set up as a UI tour that describes elements of the user interface and so reduces the cognitive load for users.

The biggest advantage of stepped contextual help is that users get a series of processable pieces of information they need when they need it. Good stepped contextual help (as shown in Figures 1 and 2) is also strongly topic oriented and respects the principles of minimalism to help a user complete a single task. Most of the time contextual help is displayed as a tooltip or speech bubble, which makes it feel as if the user is being assisted by the software itself.

Online banking interfaces, Google, Facebook, and many others follow this trend.

FIGURE 1: CONTEXTUAL HELP ON YOUTUBE

FIGURE 2: CONTEXTUAL HELP ON PAYPAL

OUR APPROACH BASED ON DITA PRINCIPLES

Contextual help has not reached its full potential. Most tours are more like a “Getting Started” guide that users go through once, at best, and then are left alone with further questions.

We wanted to create something that follows the way people actually perform actions online.
At Pronovix, we’ve been somewhat obsessed with DITA in the past few years. So when we developed WalkHub, a step-by-step, interactive tutorial and documentation system, we wanted it to combine the best practices from the technical writer community, DITA, and online UX design. We wanted a technical writer to be able to use WalkHub to create documentation deliverables that are reusable and that would fit with the other documentation developed by a project’s documentation team.

WalkHub—the repository and community hub where authors can collaborate on Walkthroughs—creates a framework for integrated tutorials that guide users through an interface, one step at a time. Just like a GPS in a car enables you to reach your destination without first having to learn the road you want to take, Walkthroughs give people just enough information at just the right time so that they can successfully complete tasks faster with much less effort.

The Walkthrough technology uses the Selenium testing format, an open standard widely used for regression testing in browser-based software projects. WalkHub itself is built on top of Drupal, an open source content management system (CMS) that has already achieved considerable traction in the enterprise world. For the actual player, we currently use Joyride.JS, an open source JavaScript library that already allows developers to create stepped contextual help (UX tours). These three components together with the features we’ve developed to integrate them and enable collaboration form an open source product that could be integrated into any organization’s documentation system.

**Topic-Orientation**

Both DITA and WalkHub are organized around the notion of the topic, defined as a unit of information that describes a single task, concept, or reference item or a chunk of information organized around a single subject.

At present, each Walkthrough created on WalkHub is a task, guiding the user through the steps required to do something on the interface. We plan to implement dedicated reference and concept topics later.

**Minimalism**

In contrast with most other stepped contextual help solutions, by default steps, in Walkthroughs don’t have a title. The text field for content entry is kept small on purpose to encourage writers to add instructions only relevant to a specific step (see Figure 3).

- Writers should give each step a description that is a maximum of two short sentences (and ideally just one).
- Each step should only address one concept or the one action that the user can take at that point.
- A step must only reference the exact item it’s pointing to (no mention of other items on the page, other pages, and so on).
- Technical terms and jargon should be avoided unless it is fundamental to understanding the step.
- A step should not contain anything that doesn’t help the reader understand the object that the step is pointing to (no titles for instance).

**Chunking**

A Walkthrough is always the smallest possible chunk that lets the user finish a task. If a process is too complex, it’s broken down into smaller chunks. The user gets the help she needs without going through steps she’s not interested in. For example, a process for completing a user profile can be broken down into tasks like uploading an image, adding a short bio, adding contact information, and so on.

**Prerequisites**

To make it possible to dedicate Walkthroughs to a single task, we implemented a prerequisite system that tells a user when a Walkthrough requires other Walkthroughs to be completed before it can be played, for example if a Walkthrough requires the user to be logged in before playing.

**Walkthrough Sets**

Another use case is when Walkthroughs are organized around a broader topic, and many individual Walkthroughs are connected by a common theme. In this case, Walkthrough sets are used, similar to DITA maps.

**Single-Sourcing**

We condense feature demos, tests, and documentation into a single deliverable. A Walkthrough can be used to generate these three deliverables automatically. The same Walkthrough can be used for presenting the features to a customer, performing functional tests, and updating the documentation (see Figure 4).
Single-sourcing in WalkHub means that you can create a demo or documentation from a Walkthrough that the developer created for functional testing. You can use a sales demo Walkthrough created by a salesperson as a sales manual or a quick test to check functionality.

You can embed automatically generated screenshots from a Walkthrough’s steps into your more traditional documentation. Any time a new version of your web application is updated, the screenshots can be updated automatically. There is no need to manually change the documentation if the interface changes. You simply rebuild the screenshots.

**Community**

One of the most important design criteria for WalkHub was to make it as easy as possible to share and reuse Walkthroughs. To this end, we created a widget from the screenshots of a Walkthrough’s steps, another incarnation of the same Walkthrough that can be embedded in a product blog or easily shared on social networks. See Figure 5.

**Reuse**

Each Walkthrough is a series of steps independent of the actual domain it is played on. As a result, you can reuse entire Walkthroughs on any number of web sites. A good example is a set of 165 Walkthroughs we created for the core functionality of Drupal 7. These Walkthroughs can be played on any Drupal 7 website that uses the core administration theme to explain settings, content management, and basic module configuration.

**Fast and Easy**

We wanted to make our solution as simple as possible. To create a Walkthrough, you record how you click through a web site. An initial description for each step is automatically generated based on the item you clicked and the name of the item is inserted. For example if you have clicked on the ‘About us’ menu item, the generated description will say “Click on ‘About us’”. Then you can update step descriptions inline while playing the Walkthrough, so there is no need to write a step-by-step procedure beforehand. To share a Walkthrough, you copy and paste a link. We know that Walkthrough creators—just like users—want to get things done quickly too.

**Summary**

This article reviews the evolution of online help and investigates the issues users and technical writers face in online communication. Intense cognitive load, frequent updates, new ways of collaboration on software required new approaches in documenting online interfaces.

We found that DITA principles can build a strong foundation for a new tool that follows the way people actually perform tasks online. To this end, we developed WalkHub, an open source, step-by-step, interactive tutorial and documentation tool with DITA concepts in mind. [1]