

User Task Flows

bringing the user into the development process

More often than not, the needs and expectations of the user are present only in the abstract when design and development of an application are being discussed. Use cases tend to reflect system interaction, and the domain knowledge and expectations of the user are generally either not documented or not documented with a relevant level of detail. Software ends up being developed before the user interaction and user models are taken into consideration.

This lack of representation results in a user interface reflecting more of the back-end framework than the front-end needs of the user. Too often, it's the user who ends up having to figure out how the software can work for them, rather than the software knowing how the user works or behaving in deference to their workflow.

Introducing task flows into your software development lifecycle, however, can avoid some of these pitfalls and benefit not only the end product but the development process as well.

What are Task Flows?

Task flows are a tool to help us think through the design before a feature is actually developed. Because task flows visually diagram an abstract idea, e.g., register, they give teams a starting point to begin discussing how a proposed feature will work at the presentation layer.

The Information Architect (IA) is generally the team member taking the lead on diagramming the user narrative. With the business requirements and user modeling taken into consideration, the IA determines the tasks needed for a specific idea and maps out how the user will accomplish each task within the application. The resulting diagram shows a series of actions the user must do in order to start and complete a task, which can be anything from uploading catalog data to generating reports to assigning rights and permissions.

Why use task flows?

They allow us to interject the user into the proposed flow of an application in order to determine if the team's conceptual model agrees with the user's conceptual model.

With diagram in hand articulating the sequence of a task, team members can perform a cognitive walkthrough, i.e., going through each step of the flow as a user would, to determine if the flow makes sense and facilitates the user's goals. This gives the team an opportunity to focus on the interaction of the proposed functionality from a user's point of view, uncover any gaps in the sequence and discuss ways the flow can be optimized - all before any actual coding has begun. At this stage of the game, change is cheap!

As the software grows in complexity and the user tasks multiply, the IA uses these diagrams to keep the interaction and flow consistent throughout the application.

Challenge:

When the features, and not the users, are the focus of a release, the end result may not be what your customers necessarily want to use.

Solution:

Incorporating task flows at the beginning of the development cycle integrates the users' needs up front.

Benefit:

The result is usability that's baked into your software from the very beginning, creating a unified user experience throughout the application.

User Task Flows

bringing the user into the development process

Where to Begin?

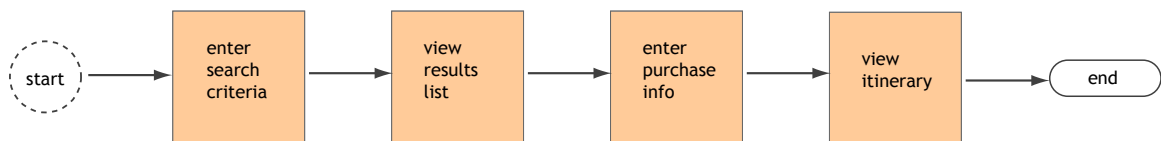
Starting with a high level concept flow, the necessary steps to initiate and complete the task are detailed and mapped out into the flow. From there, the flow is reviewed with the team and various iterations of the original diagram are created and expanded upon as more requirements become known. The end result maps out the different levels of complexity of a given task, including subtasks and parallel flows once they are identified and designed.

By placing the user in the process at this stage, the IA and the team can determine up front whether or not the final task flows meet both the business requirements and any system requirements, and still make sense to the end user.

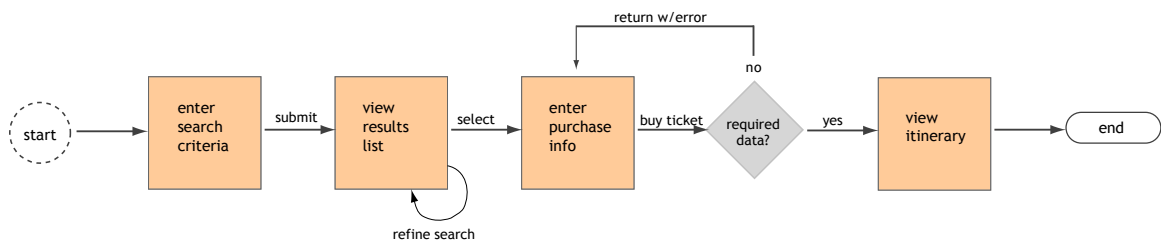
As an example, let's look at a user purchasing airline tickets. At a high level, the basic steps to complete the task are:

- Search flights
- Select flight
- Reserve flight
- View Itinerary

Which, when mapped out into a task flow, looks like this:



The team can begin using the flow to determine if the data presentation and flow sequence make sense to the user, meets the business requirements and works within the system. As details are worked out and subtasks and actions are identified, granularity is added to the diagram and the flow is revised to reflect the added complexity.



From here, the diagram is expanded as different requirements become known. In essence, each iteration of the task flow captures the dialogue of the user moving through the software, detailing both user actions and system responses.

User Task Flows

bringing the user into the development process

How do task flows benefit a project?

- **Improve product usability.**

Task flows bring the users' needs from the planning phase into the design stage of a software project. They are a means to focusing on developing software that actually aligns with the users' needs, rather than arbitrarily tacking on functionality to meet a release date. By using these task flows as touch points during development, features are designed and developed that meet the users' expectations and help them do their jobs. With this upfront effort implemented during the design phase, usability becomes a seamless part of a finished product that behaves in accordance with the users' mental model. This usability can translate into reduced calls to the support desk, a quicker adoption rate, and having future releases truly be about enhancements and not fixes.

- **Save on Development Costs.**

Working out the Workflow. Making changes after a product is released is much more costly than identifying issues up front and solving the problem before coding begins. Documenting the task flows during the design phase allows team members to walk through the tasks as a user, identifying available actions and uncovering places where the sequence doesn't match the user model. Decision points are identified and subtasks mapped out, which then gives the developer a visual of what the workflow needs to allow and how flexible it needs to be.

Modeling the Data: Creating task flows diagramming end-to-end tasks throughout the system generates a picture of how the software will be used and how the user will interact with the data. Developers will begin to identify how and where the data is needed throughout the application. This, in turn, will help them to structure and organize the data in a manner that supports the users' needs, rather than having the user modify their behavior in order to support a theoretical data model that doesn't match their reality

Helps Identify Patterns. Diagramming the task flows gives the team a wall of visuals representing various tasks throughout an entire application. Scanning the flows allows them to quickly identify recurring patterns. This, in turn, lets development begin to address creating reusable components or extending existing ones to allow the flexibility of reuse throughout the application.

- **Visually establish the larger picture.**

Documentation binds large teams together. The end-to-end flow generally isn't developed linearly by one developer or even by one team. Instead, it's broken down into features with various teams coding the stories for that iteration.

Creating end-to-end task flows is the perfect vehicle for letting a developer see the overall picture of how an entire feature works and, more importantly, how and where their piece fits

Task Flows Help:

- Improve Product Usability
- Create Reusable Components by Highlighting Patterns
- Identify and Structure the Data
- Prepare Test Scripts for QA
- Display an Overall View of the Application

User Task Flows

bringing the user into the development process

into the whole. For developers it's no longer one story in isolation, but rather a relationship of their one story with entry and exit points that define the sequence.

- **Aid in writing test plans**

Flows show the end-to-end sequence of a task, along with any decision points and subtasks. Creating a wall of task flows visually maps out a software's features and shows how the user will step through them. This information is invaluable in helping QA get a jump start on writing their test scripts by showing them the ideal sequence, alternative flow(s) and error handling before they've even seen a screen.

- **Explain the application**

All team members can benefit from task flows because they are created in a visual format that, at a glance, gives the team a basic understanding of both the overall application and the available features. They show how the features work, how they relate to each other (or don't), and their end-to-end flow. This is especially beneficial as new members are integrated into the team and need to be brought up-to-speed quickly.

External users will benefit not only from the added usability, but also because task flows can help the copywriters of the application's help text. The diagrams will let the copywriters see the overall picture of a feature before they've even used the application, allowing them to organize and write their help text accordingly. From the task flows, they can identify crucial areas in the application where it would be most beneficial to include context sensitive help and provide a more detailed explanation and/or examples.

Conclusion

Today's web applications support the editing and manipulation of data, allowing a much more robust dialogue to take place with between the software and the user. The basic concept of fetching static data from a hyperlink is expanded to include user actions such as add, delete, modify and clone, thereby allowing the user to effect some sort of permanent change. Adding in a rich interaction layer furthers the dialogue by presenting the opportunity to manipulate screen data (e.g., configuring a list view) without permanently changing the stored data.

This technology greatly expands the robustness of the user experience and blurs the distinction between desktop and web-based applications. When done well, the end result provides the user with a better tool to help them complete a task. Task flows are a means to achieving that end.

User Task Flows

bringing the user into the development process

About the Author

Alice Toth is a Senior User Experience Architect at Pathfinder Associates. She brings with her 12+ years of application design experience garnered through various Internet start-ups (and shut-downs) and consultancy. Her most recent involvement has been with designing and developing the user experience for enterprise-level web applications, including i18n and L10n requirements, used by such clients as Motorola, Nokia, FedEx and Symantec. The underlying theme of her practice is a focus on optimizing the presentation layer and work flows to achieve the ideal balance between user needs and client requirements.

About Pathfinder Associates

Pathfinder Associates builds software products for enterprises and software vendors. Pathfinder's approach integrates user experience design with agile development to improve software quality and accelerate development. This approach helps Pathfinder's clients speed intuitive and reliable software to market. Pathfinder has offices in Chicago and New York and offshore delivery centers in India and the Philippines.

Pathfinder Associates, LLC
215 W. Superior
Chicago, IL 60610
312.372.1058 x6006

For more information, please visit us at www.pathf.com

