

## CHAPTER NINE

# *Flow Charts*

### flō' chärt (n.)

Flow charts attempt to visualize a process, usually centered around a specific task or function. For web-based processes, flow charts often represent a series of screens that collect and display information to the users. Also known as flows, user flows, process charts.

Even the simplest explanations of the web acknowledge that it is far from a static medium. The relationship between users and web sites is one of movement and interaction. To a greater or lesser extent, the paths through web space must be planned, and flow charts are the designers' tool for defining those paths.

What separates a flow from a site map is that in the former, time is the defining factor. The relationships between the steps are sequential, not structural or hierarchical. While site maps capture an information structure that may or may not match the user's experience of the site, a flow chart defines a process from beginning to end. They come in all shapes and sizes, and while several conventions have emerged, there is still no single right way to do a flow chart. One reason for this is that a process may be defined at different levels—from the high-level business process to the detailed step-by-step specification of how a user creates an account on a web site.

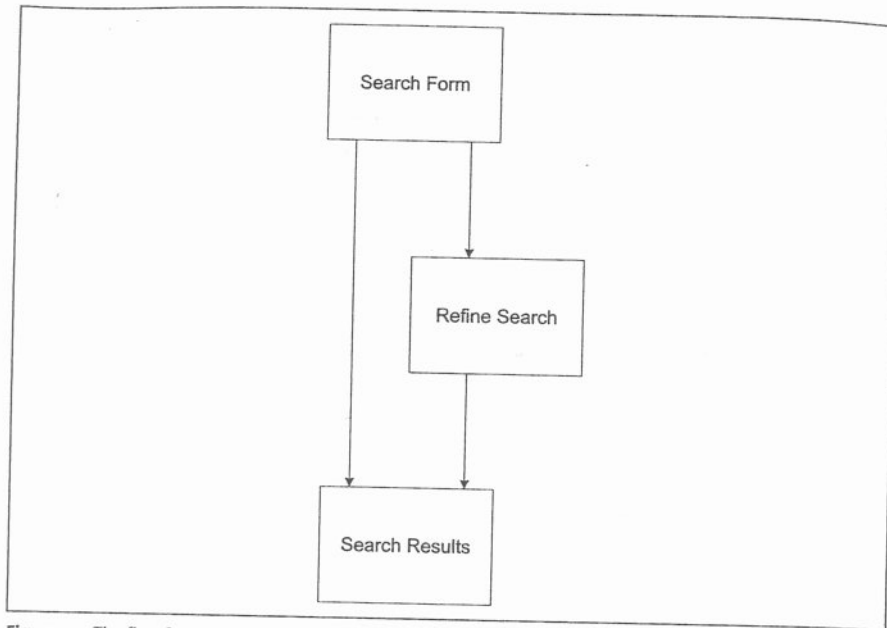


Figure 9.1 The flow for a search exemplifies the simplest flows possible—the computer requests input and provides output. This flow also shows an option to go to an advanced search screen.

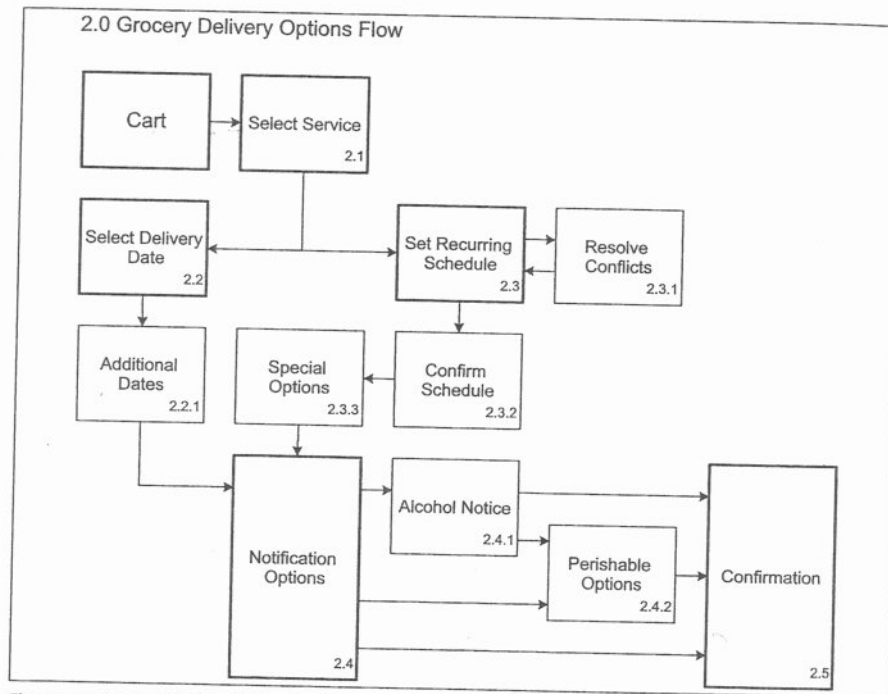


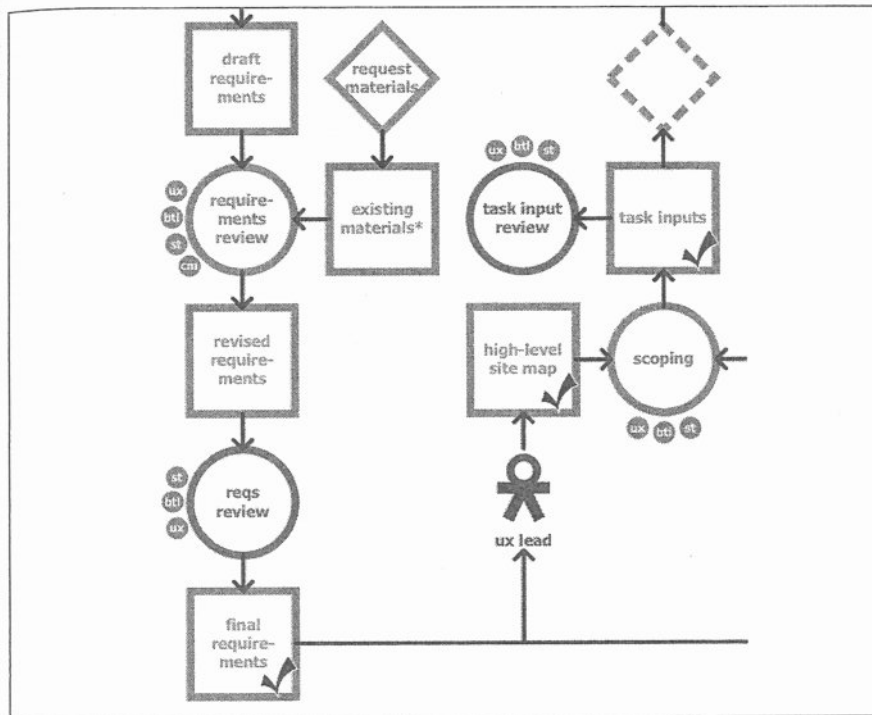
Figure 9.2 This flow is for a fictional online grocer. It shows more complexity in branching logic and relative importance in various stages of the flow.

Figure 9.3 T and arrows t shapes repr

## Flow

There are and the c people in people in charts be experienc showing

Stripped-c nected by how peop what a we of a flow c movemen ending po specific ab



**Figure 9.3** This is an excerpt from a flow describing a business process. Like a flow for a web site, it uses boxes and arrows to show different steps in the process. Instead of web pages, however, the boxes, circles, and other shapes represent actions participants take.

## Flow Charts at a Glance

There are two main uses for a flow chart: One is to show a business process and the other is to show a user experience. Business process flows describe how people interact with each other, while a user experience flow describes how people interact with a web site. This chapter will explore both kinds of flow charts because they're very similar. One thing to note about flow charts for user experience is that unless combined with wireframes, they don't do a good job of showing every aspect of the user's interaction with the site.

Stripped-down flow charts can look like site maps or concept models—shapes connected by lines. Like those other two documents, a flow chart can represent reality—how people go through an existing process—or it can be prescriptive, describing what a web site or organization should do. On the other hand, the unique layout of a flow chart is designed to communicate movement through time rather than movement through space. A site map may have a clear starting point, but not a clear ending point, and no specific path through the structure. A flow chart, however, is specific about where the user begins and ends, and has a clearly marked path.

requests input and pro-

Confirmation

2.5

logic and relative

## Flow Charts Overview

### Purpose—What are flow charts for?

Flow charts give team members a picture of how users complete specific tasks. A flow chart may not reveal all the details of the interactions, but it does offer a comprehensive view of the user experience for a particular objective. The objective can be high-level, like researching cars, or more specific like establishing a user account.

### Audience—Who uses them?

Every member of the project team will use the flow. For developers, the flow is an overview of the logic in the system, documenting each step in the process and the business rules linking them. Designers will use flow charts to plan screen designs. User flows are also a good way to give stakeholders an early glimpse into the final product.

### Scale—How much work are they?

These documents can be used at a broad level, painting an entire picture of how an organization interacts, or a microscopic level, describing how a single person accomplishes a specific task. As with any other document, the amount of work depends on the level of detail and the amount of research or planning required. Even the simplest tasks can require a complex series of business rules to accommodate every situation, which must be documented for the flow chart to have any value.

### Context—Where do they fall in the process?

User flows are flexible tools that can be used throughout the project to document different aspects of it. Business processes may constitute requirements for the system, and can be documented using a flow chart. Further into the project, the design team can use a flow chart to define how a particular function works on the site.

### Format—What do they look like?

There are many different ways to illustrate a flow. The typical approach involves using a variety of symbols to indicate different steps in the process, but some designers create a series of rough HTML pages. Linking these pages together, the designer shows—at a high level—how the flow will work. Other technologies, like Flash, offer additional approaches to describing flows. Of course, none may be more effective than a simple written narrative.

## Chall

Site ma  
represe  
erables  
is gene  
Behind  
based c  
ments,  
you're  
more c

Embed  
represe  
arrows  
may ex

The sa  
we'll h  
the flo  
is not  
represe  
dictate  
but th  
arrow:

In sho  
point  
have t  
shoul  
hotel  
cumst  
proces  
real w  
what :

The p  
tably  
that b  
design  
flow c

## Challenges

Site maps represent structures and wireframes represent screens, but flow charts represent what people do. In this way, they have a unique position in the deliverables pantheon. Sure, a flow chart can represent a series of screens, but there is generally an enormous dependency on user action to move the process along. Behind the scenes, the web site needs to make decisions about what to display based on what information the user supplies. Compared to all the other documents, flow charts most represent a conversation between user and web site. (If you're using flow charts to represent business processes, the conversation is even more complex.)

Embedded in the document is the assumption that this is an effective way to represent reality, that our processes really do look like boxes connected by arrows. And it's easy to get caught up in this assumption. At our worst, we may even become indignant when reality doesn't fit so nicely.

The sad truth is that we have little better way of talking about process, and so we'll have to settle for boxes and arrows. (Even those among us who can make the flows look especially sexy know that under the hood, every process chart is nothing more than boxes linked by arrows.) Acquiring the information to represent a process effectively and sorting out all the business rules that might dictate the various paths through the process may be the hard part day to day, but the real challenge comes from what we lose when we resort to boxes and arrows to represent reality.

In short, everyone is different and brings something different to the table. The point of a flow chart is to normalize users, smooth them out, and force them to have the same conversation with our web site every time: Every flight booking should look the same, every online gift purchase, every bill payment, and every hotel reservation. But behind every one of these actions is a person whose circumstances differ from the user who came before. As much as we'd like every process to be executed the same way every time, this just doesn't happen in the real world. Slightly varying circumstances introduce complex processes, and what appears simple on paper is in reality much more involved.

The problem with a flow chart is that these circumstances are invariably, inevitably lost. There's no way we can tell a complete story because the situations that bring people to the flow are infinite and inherently unpredictable. As designers and developers, our job is to keep the user's best interest in mind. A flow chart—as useful as it can be—can get in the way because it represents this

crazy mix of user needs as a singular, unified whole. The greatest challenge for us working on flow charts is to take a step back, look at them critically, and ask ourselves “What’s really going on here?”

## Creating User Flows

Despite the challenges, a flow chart must tell a story with the user as protagonist, and therefore must include a beginning, middle, and end. The middle is the meaty part, showing all the places the user may go before reaching the end.

Like all the deliverables in this book, flow charts are defined by three layers. The first layer includes all the essentials: Leaving one of these things out would fundamentally change the deliverable. Even the simplest flow charts have everything in the first layer. The second and third layers contain additional information providing additional levels of detail for your audiences. Whether to include them depends on your audience, your project, and your team.

### *Layer 1: Must-Haves*

Flows have a few more minimum requirements than site maps. Site maps show structure, but there’s no prescription for the type of structure, whereas flow charts have more direction. The essential elements of flow charts include more than just the steps in the process. You need to indicate starting and ending points, as well as any rules that guide the direction of the process.

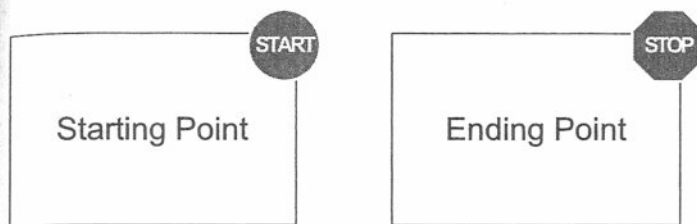
### **Anchors**

Every flow has a starting point and an ending point—that is, where users enter the process and where they conclude the process. The flow documentation should show this. There are several different ways to distinguish endpoints from the rest of the flow.

If the flow represents a modular process, the starting and ending points can be distinguished with unique symbols that represent users entering the flow from somewhere else. This matches traditional flowcharting notation, which uses a circle to indicate beginning and ending. Typically, this circle will include a unique identifier for the flow and where it returns upon conclusion.

Self-contained flows that represent a main process in the application may start on one of the site’s main pages. In this case, you can use the same symbol for

this page as other pages, with slight variation to distinguish it, thickening the line weight or changing the background color.



**Figure 9.4** One way to represent starting and ending points is to use “go” and “stop” symbols. It’s glib, but effective.

## Steps

The shortest processes consist of two steps: collect input, show output. Online search applications are the best examples of this: They ask for a keyword and display results matching the keyword.

In some cases it may be useful to distinguish the importance of various steps. Some processes, for example, have a set of required steps. Every user must move through these steps in order to complete the process. Depending on options they select, though, there may be secondary steps.

These days, a shopping cart is a shopping cart. You’ve seen one, you’ve seen them all. From the main shopping cart page, you are asked for shipping information, billing information, and a confirmation. On the other hand, users choosing international shipping, gift options, or gift certificate payment options may need to provide additional information that temporarily takes them “outside” the main flow.

If necessary, you can distinguish between the flow’s primary steps and other parts of the process by changing the size of the symbol representing a step. Larger symbols represent main steps while smaller versions represent secondary processes. I do not recommend changing the shape entirely since a step is a step, regardless of its importance in the flow.

## Paths

No doubt most stakeholders are used to the convention of lines between shapes representing paths between steps. Although it may be tempting to change the style of the line to show whether paths are required or important, use caution here. Too much variation—thickness of lines, color of lines, dotted lines, dashed

lines, double lines—can create visual noise, making it difficult to read the flow. Err on the side of simplicity and use notation rather than style to show differences in paths.

Like the space between panels in comic books, there is a lot happening in the “space” between steps, and the little lines between boxes are burdened by communicating this information. If there are conditions for moving from one step to the next, the document should reflect these conditions.

One simple way to do this is to include a special symbol on the line to show that it must meet a condition. You can have a callout to describe the condition right next to the path, or you can label the line with a number or letter and describe the condition in the margin.

### Decision points

Complex conditions may be represented by their own nodes. In other words, a decision point in a process may be represented by its own shape. Traditional flow charts use a diamond. Typically, decision points are expressed as yes-or-no questions, with yes responses taking the user in one direction and no responses in another direction.

Examples of binary decision points:

- Is the user logged in?
- Is the user placing an international order?
- Is the data supplied by the user valid?

Don't feel constrained by this binary approach. Not every decision point should be documented this way. For example, to show a logged-in state, simply include two lines coming from the starting point with one labeled “logged in” and one labeled “anonymous.” Or, you may want to reserve decision points for more complex logic and express them as system imperatives:

- Determine required taxes
- Determine access privileges
- Filter results

Try to design your flow such that path lines do not cross. Some paths make this difficult, but ultimately the diagram will be easier to read if you keep line-crossing at a minimum.

### Process

Every process talk about task, or financial account s

You may users or nu cated eno refer to th

### Project a

Besides th can help i if you're n members- things stra the names name. You how the d few pages,

### Layer 2.

What disti contain en informatio By thinkir

### Step disti

In the first the process want to sh web-based some infor mation bas kinds of sc over time.



## Process name and identifiers

Every process should have a name, if for no other reason than to make it easy to talk about. The name should be meaningful and related to the process' purpose, task, or function. Here are some good names: new user registration, log-in, edit account settings, establish new wish list.

You may want to give the process some other identifier, such as a series of letters or numbers, like A1 or 6.2. Your numbering scheme should be sophisticated enough to distinguish the different steps in the flow, so that 5.1 and 5.2 refer to the first two steps in flow 5.

## Project and bookkeeping information

Besides the name of the process, there are several other pieces of information that can help identify your document. This information may seem extraneous, even if you're not using a lot of documentation, but it can help stakeholders and team members—who have a lot more on their minds than your flow chart—keep things straight. Your flow chart should make room for the name of the project, the names of contributors, date and version number, and perhaps the client's name. You may also want to leave room for a version history—a description of how the document has changed over time. There are more details on this in a few pages, but I wanted to tell you now so you won't be caught by surprise later.

## *Layer 2: Further Details*

What distinguishes the second layer from the first is level of detail. Layer 1 should contain enough information to build the application, but the second layer of information can provide further detail to prevent confusion later in the process. By thinking about these distinctions now, you and your team will save time later.

## Step distinctions

In the first layer of a flow chart you might distinguish between major steps in the process and minor steps. There are, however, further distinctions you might want to show by varying the look of the symbol for steps. For example, many web-based processes are a combination of give and take—the web site requests some information and provides a response, and the user offers additional information based on the response. Your flow can visually distinguish these different kinds of screens to show how this “conversation” between user and site emerges over time.

## Step details: Who, what, where, when, why

The bare essentials of a step include a rectangle and a label, but there's much more to a step in a process than its name. There are any number of questions you might answer about a step in the process: Who performs it? What are the inputs and outputs? When does the step occur? And why is the step important?

In some cases, this information is part of the design, perhaps because the process is new. For example, a complex process might include error-checking routines, subprocesses that communicate to users whether they've provided incorrect information or committed some other error. Since these kinds of diversions from the normal process can be confusing and very detailed, you might include the error messages themselves in the flow.

Even if you do not need to specify this information as part of the design, it can provide contextual cues to stakeholders, helping them understand how the step fits in with their understanding of the organization. By relating specific people in the organization with steps in the process, you make the steps more meaningful by associating them with something recognizable.

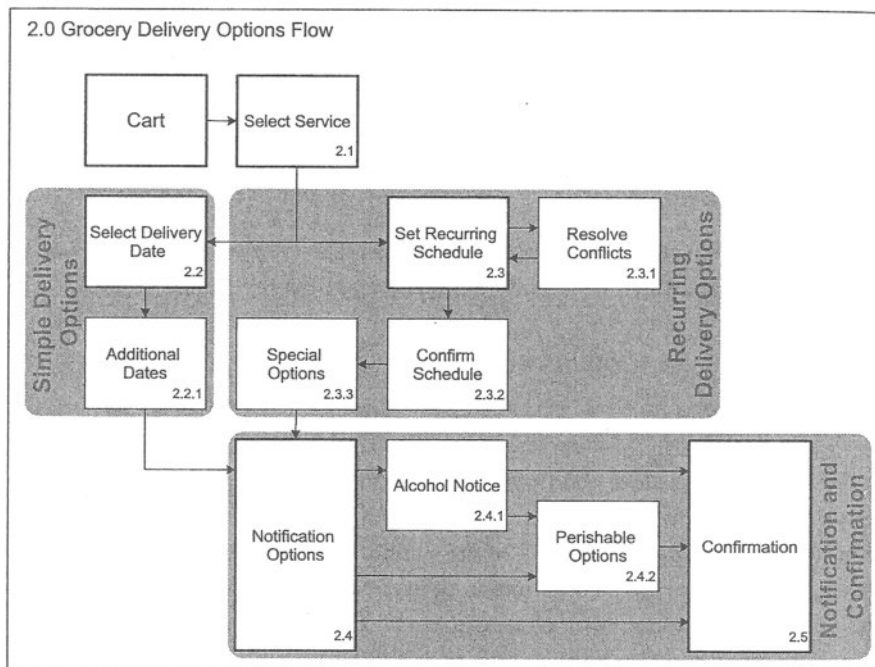


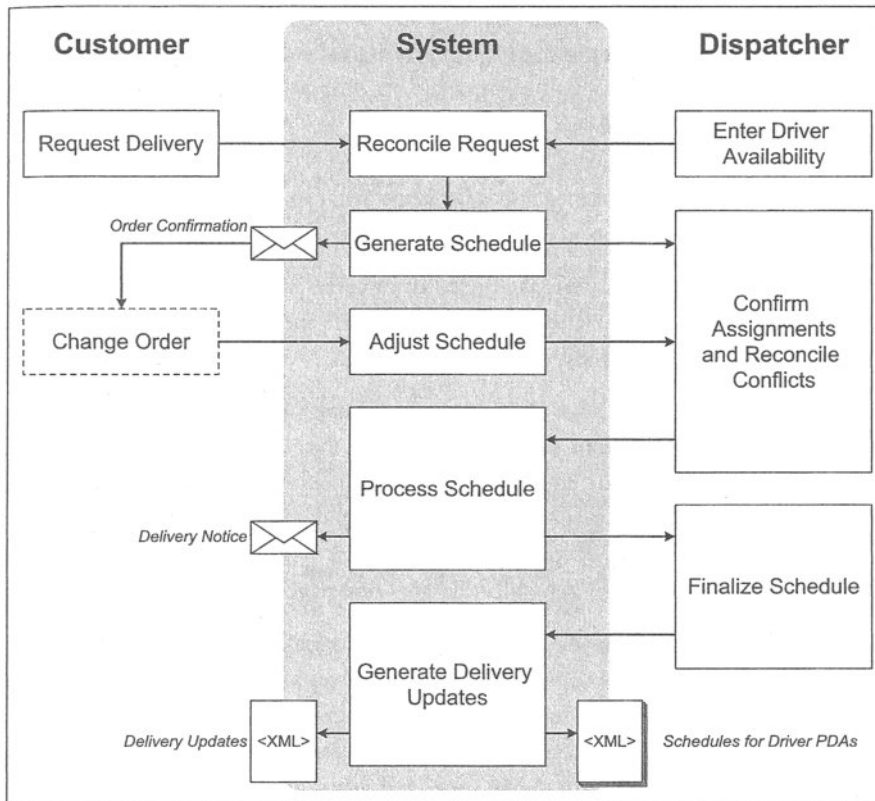
Figure 9.5 This complex process has been grouped into related functional areas, shown by the labeled, shaded backgrounds. The groupings make it easy to talk about different parts of the process.



## Step groupings

Your process may consist of related groups of steps. These relationships may or may not be evident from the paths between them. Even if the paths do clearly show groups of related steps, an additional visual device may help clarify those relationships. Grouping related steps allows you to apply a name, which can make talking about the process easier.

In one type of grouping typically associated with flow charts, steps are organized not only in sequence but also by who is responsible for them. The template for this type of process includes “swim lanes” where each person or role is assigned a different lane, and the steps owned by that person appear in his or her lane. For web-based processes, one swim lane might represent the system itself, showing how the computer, in effect, contributes to the process.



**Figure 9.6** This mock-up shows how a process might be organized around “swim lanes” which represent different people responsible for different parts of the process. In this case, the system—represented by the shaded swim lane—also takes some of the responsibilities in the process. Shapes overlapping two swim lanes can show joint responsibilities or, in this case, how the system communicates with the humans involved in the process.

## Error paths

Error scenarios describe what happens to users when they commit an error—neglecting to enter information in a required field, for example, or supplying inconsistent information. In some cases it is worthwhile to document these on the main flow and in others to separate it.

Documenting error scenarios is difficult because some interactions with users present many opportunities for users to commit errors—requesting payment information, for example. Illustrating all these possible errors requires a lot of detail. In complex processes, it may be difficult to capture all this information in one place and makes more sense to document each scenario separately.

Flows may not be a good place to document error scenarios at all. Wireframes, for example, are a better place to capture error scenarios because you can associate them with specific areas of the page. Some error scenarios are repetitive, and illustrating those paths on the main flow may make it difficult to read.

## Variations on each step

Early web-based applications were easy to document because the “page model” was well entrenched; that is, web sites were built and programmed as if they were a series of pages. As technologies advance, however, it is more difficult to conceive of web applications as a series of discrete pages. Instead, a single “page” may have many variations depending on the mode or state, which could change depending on business rules or user actions.

For example, the landing page of an application may vary depending on whether the user is logged in or not. If the user is logged in, the landing page may contain different fields, even though it is conceptually the first page of the application. The increasing complexity of web applications creates new challenges for documentation. Representing these page variations in a flow can be difficult where conventions generally dictate “one page, one rectangle.”

One technique would be to create separate flows for different scenarios. One flow would address the user’s “logged in” experience, and another the “anonymous” experience. With this technique, you might want to also show which pages are “shared” between scenarios. Another technique is to represent each page variation separately on the same flow, and group them visually. This can demonstrate that conceptually this group of variations is the same page—the user will never see more than one at a time—but its display varies depending on state

or mod  
to docu  
appear

## Layer

Layers  
to inclu  
entire p

## Trigge

For the  
contair  
in the p  
ing the  
place. I  
tem rec  
the trig  
you can  
By the  
thing s  
The flo  
the pro

## Scena

Like tr  
Scenar  
just wh  
to prov  
cases v  
differe  
users t  
inform  
proces  
custom  
the “n  
on the

or mode. Whatever technique you use to show page variations, it is important to document the scenarios, the circumstances in which one page variation will appear over another.

### *Layer 3: Further Context*

Layers 1 and 2 focus on the flow itself and details about the steps. If you decide to include additional information, you might focus on the context behind the entire process.

#### **Triggers**

For the purposes of web design, processes are typically conceived as self-contained mechanisms that accept inputs and produce an output. The steps in the process account for the transformations to the input necessary for achieving the output, but none of this addresses how the process happens in the first place. In some cases, the triggers are obvious—a user clicks “checkout,” a system receives a search query, or some such distinct action—but in other cases, the trigger requires some explanation. Near the starting point of the process, you can put a description of what conditions are necessary to start the process. By the same token, just because a process concludes does not mean that everything stops. The process may in turn trigger another process, or set of processes. The flow chart’s stopping point can include a description of what happens when the process ends.

#### **Scenarios**

Like triggers, scenarios provide context and background for the process. Scenarios are broader than triggers, telling a story about the overall process, not just where it begins or how it ends. Sometimes the title of the process is enough to provide context—“new user flow,” for example—but it may not be clear in cases where the flow depends on a number of outside factors, or includes many different functions. For example, imagine an online banking site that allows users to sign up for a variety of accounts. Due to the complexity of the account information, each type of account has a slightly different flow. What’s more, the process is essentially the same for a completely new customer and an existing customer who just wants to add a new type of account. You can just call this the “new accounts” flow, but you may need to spell out some of these nuances on the document itself.

## *Going with the Flow: The Basics*

On the surface, flow charts show the user's progress from one end of the system to the other, but, like every deliverable, they serve a deeper purpose in the project as well. The purpose of a flow chart is tied to its situation and audience. Since flow charts represent an abstract stage of the design and can appear just about anywhere in the process, they can serve a variety of purposes.

### **Purpose**

As a design document, a flow chart's main purpose is to capture part of the user experience. But you can also create a flow chart to capture your understanding of a client's business process, separate from the user experience. For example, the checkout process on a commercial site is different from the internal business process for fulfilling a customer's order. As with any deliverable, the purpose of your flow chart will dictate, to some extent, the kind of information it includes.

When documenting the user experience, flow charts generally show a series of screens. Each step in the process is one screen, which is generally a form for entering information or a response based on user input. The arrows represent different paths depending on users' responses to the questions on the screens. A flow chart describing user experience should also include technical information for the developers describing the rules applied to the information supplied by the user. A simple example would be displaying different screens to users in a checkout process depending on whether they've selected the gift option. These are rules that operate behind the scenes, but have an important impact on the user experience.

Alternatively, you can create flow charts to represent business processes, which show the steps employees must follow to accomplish a task. This kind of flow chart can be useful for understanding how the web site should support the tasks performed by people in the organization. For example, you might use a flow chart to document an internal fulfillment process, where different workers have responsibilities for capturing orders, preparing the orders, packaging the orders, labeling them, and sending them off. Understanding this process will help you design the flow for the online system because you will know what kind of information you need to collect from users in the checkout process of a commercial web site.

Documenting business processes is even more important when you're working on an internal web site, which might support these internal processes. For

example, you might capture the editorial process for creating and publishing a specific type of document in order to build a content management system to support it. In this kind of flow chart, technical rules might not be as relevant immediately. Instead, you'll focus on ensuring you've captured all the key decisions in a process, and the important participants.

### Timing

Early in the process, the flow chart may have less functional detail and show more business context. Flow charts developed later in the design process will emphasize functional decisions.

One way to make sure your flow chart is appropriate to its position in the project is to make a list of all the inputs it will include. By identifying what you have to work with—specific requirements, results from usability tests, direction gleaned from personas—you can be sure you've accounted for existing work on the project. This list should also indicate what will come after the document. A detailed flow chart could lead right into development, or into further design activities, or prototyping. Understanding what comes next can help shape the document.

Early in the design process, you may not have all the functional details decided, in which case the flow chart serves as a method for encouraging discussion and brainstorming. Alongside the flow, you can include any questions you might have about the process or the functionality as points of discussion.

Later in the process, when you've got most of the functional issues ironed out, the flow chart must capture all the details that define the flow. With this kind of document, there must be strong connections to other design deliverables, like wireframes, to pull all the functional details together. References to earlier documents will help ground design decisions. At late stages in the design process, this deliverable serves as documentation for the system, and the details must be fully realized to support development efforts.

### Audience

There are three primary audiences for any design document, and knowing the primary audience will help you decide what kind of information to include as well as how to spin that information.

For stakeholders, offer a basic description of the flow:

- Users move from the shopping cart page to the shipping information page. The shopping cart page will allow them to make changes to the quantities of products. The shipping information page will ask them for a location to ship their product.

For developers, emphasize business rules because these will have to be coded:

- Moving from shopping cart to shipping information, the system needs to check whether the user has set any quantities to zero. In this case, the system should not confirm the deletion, just assume the user wants to delete the product from her cart. No other error checking happens between the cart and the shipping information.

For designers, highlight the trigger mechanism because this is what will likely get the most visual attention:

- To move along in the process, users will indicate they want to check out from the shopping cart page. This will load the shipping information page.

Note that while all these audiences may be most interested in the contents of the pages themselves, that conversation is premature and those statements are kept to an absolute minimum.

Because of their distinct perspectives, each audience will need something different out of the flow chart. It may be difficult to accommodate all this information or to create three different versions of the diagram. By identifying your main audience, you can overcome these difficulties by focusing on the key messages for that group.

### Content development

In thinking through the situation and the audience, you may find that you still need to do a bit more planning before putting a flow down on paper. Creating an inventory of all the different kinds of information you want to include will ensure you do not forget anything in the flow. It may also help you think through some of the issues in documenting a process or user experience. The document layers listed earlier capture these elements at a high level, but you'll need to create a more specific list, tailored to your specific situation, audience, and need. You may also find that it helps you identify those elements that you do not want to include.



You might prepare this list before determining the situation, audience, and need, and then use those other lists to prioritize the elements or even eliminate some of them. Alternatively, you could create the elements list after having established the situation, audience, and need. Either way, by creating a list of elements ahead of time, you can also think about how you want to represent them visually in the flow. (Sound familiar? This technique was also recommended for site maps.)

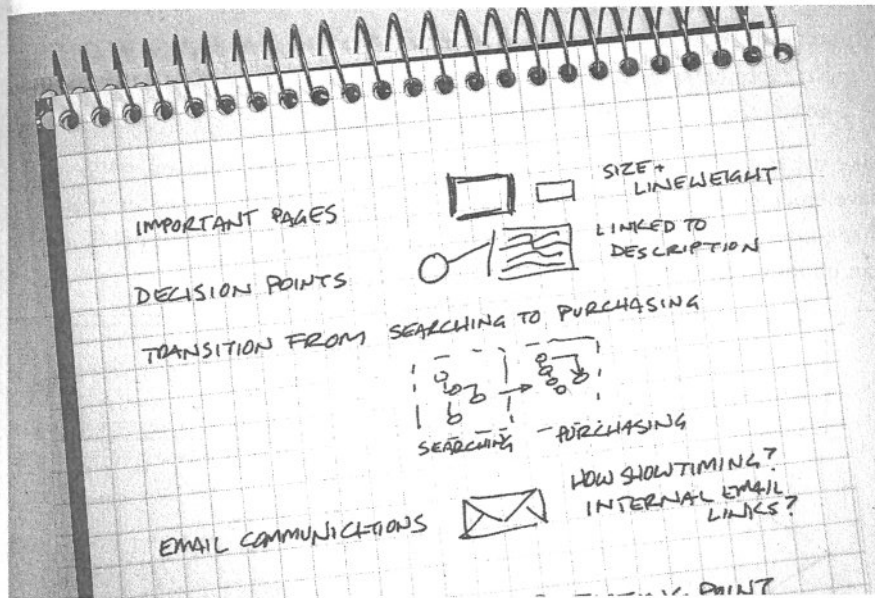


Figure 9.7 This handwritten list helps the designer get his head around all the things he wants to include in the flow. He's also started identifying how he'll represent each element visually.

Just because an element is on your list does not mean it must have a shape associated with it on the diagram. Good diagrams find ways to consolidate information, using a variety of visual formatting devices to communicate multiple dimensions.

For example, putting a lock icon inside a rectangle to represent pages requiring log-in adds a visual element to the page that might compete visually with other, more important elements. Compare the importance or relevance of a visual element to its visibility. In this case, making all logged-in pages a similar color may do a better job communicating this element because it easily distinguishes it from other pages and avoids adding clutter to the diagram.

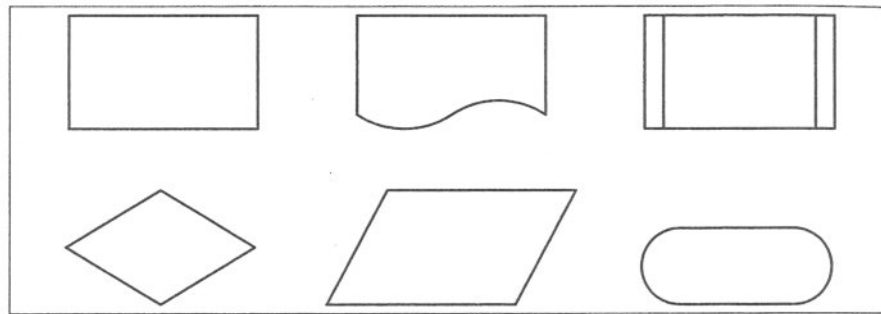
## Greasing the Wheels: Tips for Effective Flow Charts

Even if you've listed out all your content and anticipated the needs of your project and team, there are a couple more things you can do to make sure the creation of the flow chart goes smoothly.

### Identify a visual language

As suggested earlier, you should determine a visual language—a consistent set of symbols—to represent the different aspects of the flow. Consistency is crucial: If a step is represented as a circle in one place and a square in another, there should be a good reason for the distinction.

Like site maps, flow charts have a long history before the web. Conventions have emerged and evolved to pace the development of technology. Links to shape libraries produced by information architects and available for public use can be found at [www.communicatingdesign.com](http://www.communicatingdesign.com).



**Figure 9.8** Flow charts were taught in my elementary school computer science classes in the early '80s. These symbols have been used for decades to represent the internal logic of computer systems, and have since evolved to describe user experience. Even back then, I was a stickler for good documentation.

### Start with crucial landmarks

If you're struggling to represent the flow because there's too much going on, go back to the basics. Pull out the handful of steps representing the major junctions in the process and put them on paper. Once you have a shape for those, you can add additional important steps. Decide early on what's important to communicate. You may be more concerned, for example, about alternate paths or error scenarios than the details behind the main process. If this is the case, this is the kind of information you should start with, after establishing the crucial landmarks.

Figure 5  
more in

**Risk**

Despi  
your  
come

**Expr**

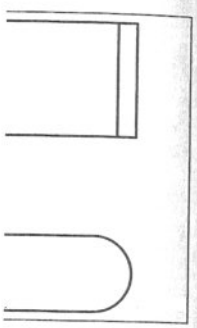
As de  
mani  
betw  
the s:  
start  
out c  
along

Perh  
and

arts  
eds of your proj-  
ke sure the cre-

a consistent set of  
ency is crucial: If  
her, there should

Conventions  
gy. Links to  
for public use



the early '80s. These  
s, and have since  
ation.

ch going on, go  
major junctions  
or those, you  
ortant to com-  
nate paths or  
is the case, this  
ng the crucial

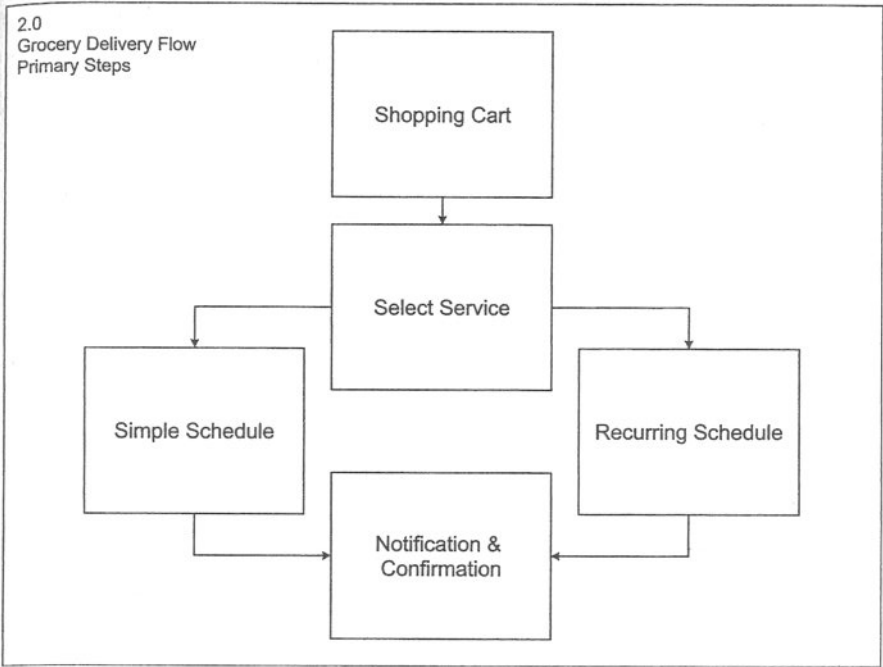


Figure 9.9 In this planning document, the designer has drawn out the main steps in the process, but has a lot more information to layer in. He's already started prioritizing his list of information to include.

### Risks

Despite all your planning, you may still run into a few problems as you develop your flow chart. Here are some ideas about how to deal with them when they come up.

#### Express what's really happening

As described in the "Challenges" section, the biggest risk with flows is dehumanizing them. Flows have a tendency to smooth over the natural variations between circumstances, giving the impression that every user will go through the same experience. Getting caught up in the details of the flow, you might start to see your users as automated repositories of information who simply spit out data when it's requested, rather than individuals who bring unique scenarios along with them.

Perhaps the best way to avoid looking at the system as nothing more than inputs and outputs is to include personas right on the flow itself. You might even add

further information for each persona, the circumstances in which they would be coming to the site, or the kinds of information they will and will not have in hand when they start the process. By superimposing information about your users over the flow itself, you stand a better chance of seeing the flow in more realistic circumstances.

### Dot the i's and cross the t's

As conscientious as you are to capture every detail, you might still be missing something. The risk here is that you'll make it to the implementation stage and, in building the system, realize that you haven't accounted for a rule or potential scenario. You'll be scrambling to come up with a solution, but won't have time to design the right one.

The catch-22 is that the best way to identify whether you've missed a crucial detail is to use the system—and, of course, you can't use the system until you've designed it. Perhaps the best way around this conundrum is to recognize that there's no way to hash out every detail just by building a flow chart. A flow chart is a starting point for implementation, and it needs to cooperate with other design activities to ensure that every aspect of the user experience is documented. But the only way to find out if you've missed something is to build the web site and test it.

That said, there are still a few things you can do to make sure you've got as complete a flow as possible. First, develop a set of criteria for each step in the process, a litmus test to help you think through all the implications. This can vary from project to project, but it might include things like expressing in plain language the purpose or role of the step—for example, to gather shipping information from the end user. You can then ask yourself a series of questions about that task: What information is necessary for this task? When the system is done with this task, what will it do with each piece of information? Minute questions like these can uncover important details. With this technique, you're testing the comprehensiveness of the flow chart by imagining the actual system.

Another technique is to recruit another pair of eyes, even if they don't belong to a user. Tapping another designer or analyst or developer and walking him or her through the process will give you a different perspective, one that can shine light on details you might have missed.

Perhaps the best way to avoid losing sight of the nuances is to do a bit of role-playing. This technique is easier if you've already prepared personas and have some direction on who will be visiting the site. Stepping through the flow, put

yourself  
tion the  
Think  
of user:  
same p  
spectiv

As this  
is inevi  
through  
strategy  
second  
try to g  
potenti

### Keep t

Flow c  
design,  
you'll h  
to do s  
necessa  
ing it u  
there is  
update:  
on the  
of the f

Your al  
to build  
that m:  
add or

One w  
above i  
updati  
these e  
find th  
won't r  
perhap  
docum

yourself in the shoes of one of your users and think about the kinds of information the site is requesting, and the kinds of information the user has on hand. Think about the order in which the site asks for information, and the likelihood of users being able to answer questions in that order. Putting yourself in the same position as the users will get you to experience the flow from their perspective, and perhaps allow you to envision the flow as they do.

As this book has said elsewhere, this problem of overgeneralizing your users is inevitable. It's just not possible for one person or a group of people to think through every circumstance and account for every detail. The best mitigation strategy is to build time into the schedule to account for these revelations. The second best mitigation strategy is to set the team's expectations: They need to try to get every aspect of the flow right, but the only way to identify all the potential issues with the system is to let actual people use it.

### Keep the document up to date

Flow charts are hard to keep up to date because in the natural evolution of a design, flow charts may be quickly left in the dust. Before worrying about how you'll keep your flow chart up to date, you should decide whether it's essential to do so. If you're using a flow chart to kick-start the design process, but not necessarily as system documentation, you may decide that going back and keeping it up to date after changing the flow may not be worthwhile. If you decide there is some value to maintaining it, you should also decide how detailed your updates need to be. Do you need to capture every nuance? Can you just focus on the big changes? Answers to these questions will be determined by the role of the flow chart in your process, and how it relates to other documents.

Your ability to maintain a flow chart depends somewhat on the tool you've used to build it, but more on how you've constructed the document. A dense layout that makes use of every inch of the page can create problems when you need to add or change information.

One way to deal with this is to look through your list of elements (described above in "Content Development") and determine which ones will require updating. In constructing the document, you can make sure to leave room for these elements. Additionally, you can prioritize these elements, so that as you find the flow becoming too dense, you can eliminate information that you won't need to keep up to date. (Not to say this information isn't important, but perhaps you can capture it elsewhere, especially if maintenance is crucial to the document.)

Finally, always leave room for a version history. Generally, putting the version history on the same page as the flow itself is effective because it keeps the evolution of the document top of mind for other people on the team.

## Presenting Flow Charts

Presenting flow charts to your audience can be challenging because although they represent something concrete—the interaction between system and user—they do so in an abstract way. Certain details like screen elements are necessarily absent because it is important to plan the interaction before tying down the interface. At the same time, it's much easier to talk about the user experience with an interface in front of you—people tend to provide better feedback on a wireframe or screen design. To help your audience understand the flow without having to resort to screen mock-ups, there are a few different presentation styles you can use.

### *Meeting Purposes*

Your presentation style depends on the purpose of the meeting, and your agenda should be driven by what you want to get out of the meeting. For example, if you just want to provide a progress report, you may not need to go into great detail. On the other hand, if you're looking for buy-in, you may need to get your hands a little dirtier.

### **Introduction and overview**

If you're not looking for any significant feedback on the flow chart, you can hold a short overview meeting. In this meeting, you might hit the high points of the flow chart, perhaps including the factors that drove the design decisions. In a short overview meeting, you may want to keep the deliverable itself simple. Too many details could lead to questions you are not ready to answer. (On the other hand, this is also a good opportunity to educate stakeholders and others about the complexity of these processes.)

In your overview, describe the important screens in the flow and the information the user must provide on those screens. Focus on the screens in the main process—the one that helps users complete the desired task. If there are any important supporting screens—those that appear when the user makes an error, for example—run through those next. You don't need to hit every screen and

every  
of the  
what  
comp

Your  
struct  
objec  
ing o  
have  
at a d

### **Feed**

Going  
need  
Brain  
tabs o  
questi

A bra  
ideas :  
useful  
Boilir  
or fea

When  
a sprin  
mum  
looks  
into tl  
Figure  
means

You n  
meetin  
those :  
into tl  
the flo  
brains  
the sit

every nuance, just those that help meeting participants get a sense of the shape of the flow. To bookend the discussion of the flow chart, describe the triggers—what would cause users to get to this flow—and what happens when the user completes the flow, or how the system and user have “changed” as a result of it.

Your web site may include many different flows. You can follow the same structure for each of the flows. Keep in mind that in an overview meeting, the object is for the meeting participants to come away with a general understanding of the flows in a system, in other words, how the high-level requirements have been translated to features, not how the user will interact with the system at a detailed level.

### Feedback and brainstorming

Going into this kind of meeting, it's useful to have a list of all the things you need help with—all the problems that need solving and unanswered questions. Brainstorming meetings are Petri dishes for tangents and distractions, so keep tabs on your progress by ensuring you're getting answers to all your outstanding questions.

A brainstorming meeting may require stimuli—inspiration to help generate ideas and provide boundaries. If you've documented requirements, these can be useful to include in the meeting, but not in the form of an enormous document. Boiling the requirements down to a handful of bullets for a particular function or feature gives meeting participants enough to go on for brainstorming.

When you need help fleshing out the basic design, the deliverable can serve as a springboard for discussion. In the case of flows, keeping the polish to a minimum can help: Participants may be unwilling to contribute if the deliverable looks like it contains fully realized ideas. You can also integrate your questions into the document itself, showing specifically where you need clarification. Figure 9.9 shows a high-level flow, with just the key areas, that can serve as a means for generating discussion.

You may also want to go in with nothing besides an agenda. You can spend the meeting capturing ideas on a whiteboard, then use the deliverable to document those ideas. Regardless of your approach, it is during this meeting that you dig into the details, and spell out the business rules for the flow. Talking through the flows of your site can help identify crucial functional details, and through brainstorming you can determine the most appropriate rules and behaviors for the site.

## Buy-in and approval

You may need buy-in on the whole soup-to-nuts process, or just on the germ of an idea. In either case, you need to demonstrate that you've thought through all the implications of the design. Select a presentation style that's going to demonstrate that you've addressed all the different aspects of the process and have accounted for all exceptions. If you show up with a radical idea without having understood its impact on the rest of the system or the business, no doubt you'll be sent back to the drawing board.

Perhaps the one exception to this is when the meeting participants have been active contributors to the design process—they've followed the progress of the design all along the way. On the other hand, it may be irresponsible to assume that your stakeholders have the exact same ideas that you do: Even when you think your stakeholders are on board, they may have a different conception of the user experience, despite their continuous involvement.

In reality, your buy-in meeting can't stand alone. If you're seeking approval on a design or concept, your best bet is to have introduced and discussed the idea before asking for your stakeholders' blessing. The content of a buy-in meeting should be simple: You need to remind the meeting participants about the background and purpose of the flow, point out the highlights, show how you've addressed concerns that they've raised previously, and show how you're mitigating risks. (It sounds so easy when written out like that, doesn't it?) Like an overview meeting, you don't need to dig into the details, provided the rest of the meeting participants have been exposed to them before, or have been involved in the design.

## *Meeting Structures*

Because flows are little stories in and of themselves, it's easy to structure a meeting around them. By far the most effective meeting structure is the narrative form, in which you walk participants through the flow as if you were a user of the web site. There are a number of other approaches you can use, however, and you should select the structure that best suits your purpose and audience.

### **Narrative meeting structure**

In a narrative approach, you tell a story about someone actually using the site. Real-world scenarios can make the interaction come to life. This works even better if you've previously introduced user personas. By employing the user personas to tell the stories, you are relating a new deliverable to one the



audience is already familiar with. This creates continuity and demonstrates that the flow was not created in a vacuum.

To use this style of presentation, you must identify the scenarios you will walk through. It may be tempting to walk through every scenario, but it's better to choose a number of scenarios based on your audience. If your primary audience is stakeholders and your goal is simply buy-in, use only a few scenarios that demonstrate the main features. For designers and developers, who will need other issues highlighted for them, select scenarios that have the greatest impact on their work.

Having selected the appropriate scenarios, you should elaborate on the details for each one. For each step in the process, make sure you're able to explain what information users must provide and the kinds of responses they will get. You'll want to describe what triggers each step in the process and what happens when the user has completed the process. At the same time, you need to avoid explicit references to screen elements—like “Joe clicks on Next”—because you don't want to create preconceived notions of what the screens will look like.

Your agenda for this kind of presentation will vary depending on the purpose of the meeting. For a simple overview, you can use this generic agenda:

- Introduction (includes project goal reminders, purpose of this meeting)
- Our Users (recap the user profiles)
- Flow Overview (identify three or four factors affecting the design of the flow)
- Scenario Overviews (before detailing each flow, give a two-sentence description of each one, ensuring you identify the user's goal for each one)
- Scenario 1: The basic scenario
- Scenario 2: An error scenario
- Scenario 3: An advanced scenario
- Other Possible Scenarios
- Next Steps

If you're holding a meeting to get approval on a draft flow, this agenda will work as well, though you may need to run through additional scenarios.

If you're doing a brainstorming meeting, the agenda should also include places to actually do the brainstorming. Narrative presentations are useful for brainstorming because you can indicate what parts of the scenario need fleshing out. First provide

a quick overview of each scenario, indicating which parts require additional work, then go back to the beginning and work through it step by step, as a group.

### Thematic meeting structure

Another way to structure your meeting is to select a theme and show how the flow supports it. Unlike the narrative structure, which attempts to imagine the user experience through pantomime, the thematic structure emphasizes design decisions centered around a key theme.

You can probably think of any number of themes for your flows, but here are a few to get you started.

- **The business objective:** By organizing your presentation around a set of business objectives, you describe how a flow supports particular goals. For example, if one of the goals of the web design project is to double new customer registrations, you can show how the flow has been designed with this objective in mind.
- **The user need:** In conducting user research, you may have identified broad needs from the users, like “make it easy to move from one function to the next” or “I need to share my information across different accounts.” Whatever the need, you can show how the flow supports this. As with a business objective, making a user need the central theme of your presentation means you do not have to dig into every detail of the flow, but can instead simply point out the elements of the flow that support that need.
- **Technical requirements:** Far be it from me to advocate designing a system around technical requirements, but this message may be important to your stakeholders. In these cases, there is usually some overwhelming technical consideration that directly impacts the flow. Your theme might be something like, “The legacy system has a specific definition for a user account, so we designed the new flow to accommodate that definition.” This isn’t to say that the technical requirements *drove* the design. Instead, you’re using the technical requirements to frame the conversation about the design.
- **Problem clarification:** Certain problems may seem intractable to clients or team members, and a flow can be an effective way to provide a solution or at least clarify the problem. For example, suppose your client says, “We have lots of disparate systems that all do the same kind of thing. Can we consolidate them?” Flow charts can be a useful tool for documenting each system, or documenting what the systems have in common, and then structuring the conversation to determine an answer to the question.

• Org  
niza  
cust  
mar  
You  
side

None  
design  
the fin  
depend  
makes

Inven

A mor  
and go  
less ou  
minut

To pla  
any of  
to exp  
go fir

For ea  
main  
detail

Losi

There  
one h  
the fl  
ability  
up be  
ding

Mak

As cc  
abstra

- **Organizational need:** Every online flow has some impact on the organization. As a new commercial channel, the flow impacts fulfillment and customer service. As a supplement to an internal business process—records management, for example—the online flow changes how people do business. You can use this impact as a central theme for educating people on the client side about how their jobs will be changing.

None of these meeting agendas advocate a non-user-centered approach to design. The design process can be separate from the story you tell to describe the final design, the outcome. The type of theme you select for your meeting depends on the participants: They need you to put the flow into a context that makes sense to them.

### Inventory structure

A more straightforward approach is to simply create an inventory of the flows and go through them in some hierarchical order. This order may be meaningless outside the context of your meeting, but it provides a structure for the 60 minutes you're sitting with other team members.

To plan this kind of meeting, simply make a list of all the flows, and identify if any of them have subflows. You can put them in the order that users are likely to experience them, or by risk—the flows that will require the most discussion go first—or by any other criteria.

For each flow, identify the main things that users do over its course, and the main screens they will hit. Once the inventory is in place, you can dig into the details of each step in the process.

### *Losing the Flow: When Meetings Go Off Course*

There are two main obstacles to a flow chart meeting that could derail it. On the one hand, your participants may struggle to picture the user experience because the flow chart is very abstract. The quality of feedback depends on participants' ability to envision the process, and if they can't do that, your meeting could end up being a waste of time. You want your flow to generate lots of feedback, shedding light in places you and the team hadn't considered before.

### Make the abstract concrete

As concrete as the ideas in your flow chart may be, the document itself is an abstraction. Your meeting participants may not be able to offer constructive

feedback because they can't picture the user experience. Knowing whether this is a risk or not depends on how well you know your clients. If you and your clients can comfortably have back-of-the-envelope kinds of brainstorming sessions, they may be prepared to deal with the abstraction of flow charts.

If you don't know your client well enough to know whether the flows will be meaningless to them, it will become painfully obvious in the meeting. They might complain that they want to see screen designs or that they can't picture it. Worse yet, they might sit there, nodding and smiling as if they understand everything about the flow.

If you're quick on your feet, you can change your approach in the meeting, presenting the ideas in a different way. For example, it might be best to put the documents aside and build the flows progressively on a whiteboard. This way, the client can focus on one thing at a time, and ask questions along the way without the distraction of the documentation.

On the other hand, you may realize that the clients' attitude suggests that you won't get anywhere in the current meeting. In that case, there's no harm in calling time out and ending the meeting. There's no point in belaboring a document if it's not getting you or the project team what you need. The lack of feedback may have an impact on the project schedule—the input you were supposed to get on the flows will have to wait until later in the process, creating some risk.

Knowing the disposition of your client may lead to you skip sharing the flows with them altogether—not a bad strategy if doing so would waste everyone's time. Whether you need to do flows at all is completely up to you and your team. Your development team may find value in seeing flows early in the process so they can begin planning the system architecture. Flows give designers a sense of the scope of the design, so they may be eager audiences as well.

### Stay flexible

Like many other design documents, flow charts can open your team's eyes. Their exposure to the user experience, even in abstract form, can trigger ideas that did not occur to them earlier in the project process. Even though user flows are not very concrete, they may be more tangible and closer to the final product than anything that came before.

Imagine needing to design a checkout process for a commercial web site. The requirements called for some high-level functions, like a shopping cart, capturing shipping and billing information, calculating tax and shipping

charges  
team to  
of e-co  
the flo  
orders-  
someth  
need to  
  
These  
ing ne  
to see  
requir  
ment,  
uses r  
  
The b  
gathe:  
and b  
flows  
proce  
tiona  
slight  
creep  
push  
is no  
chan

### Flc

Like  
stan

### Flo

Lik  
mer  
use:  
the  
can

charges, confirming the order, and placing the order. Perhaps you and the team took the user experience for granted—after all, with more than ten years of e-commerce, the checkout process has been refined over time. You build out the flow, and when you show it to your clients they bring up the issue of back orders—products that are not in stock, but that they’ll be getting soon. It wasn’t something that came up during the requirements process, but it’s something you need to accommodate.

These situations can be difficult because clients may not recognize why throwing new requirements into the mix at this stage can be risky. It can be difficult to see that a single new requirement can have a cascading effect, affecting other requirements and design efforts. The other issue is that the requirements document, for better or worse, drives project planning. Everyone on the project team uses requirements to estimate how long it will take them to do their part.

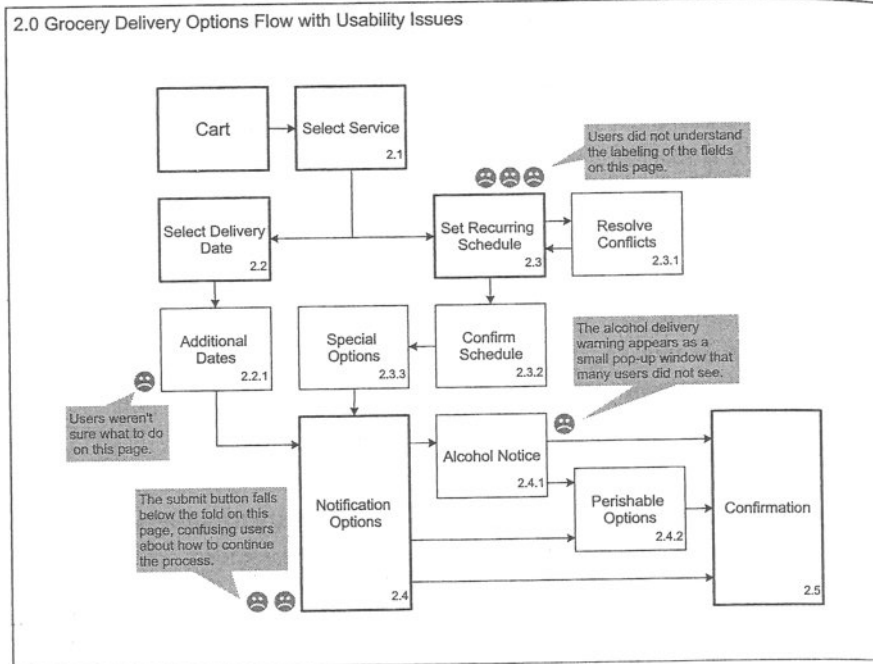
The best cure for this risk is to assume it will happen—that your requirement-gathering efforts do not end when you hand in the requirements document—and build extra time into the design process. This strategy isn’t unique to user flows, of course. There are countless opportunities along the course of a design process for new requirements to arise. People in the computer business affectionately call this “scope creep”—when the boundaries around the project shift slightly, but perceptibly. Your project should have a general strategy for scope creep, one that allows your project manager (or whoever’s running things) to push back or take the requirements away for further analysis. The key here is not to promise anything immediately because even the seemingly smallest change can have a profound affect on the scope of your project.

## Flow Charts in Context

Like site maps, flow charts are flexible and can be used in a variety of circumstances and in conjunction with nearly every other kind of document.

### Flow charts and user needs documents

Like other design documentation, flow charts can supplement user needs documentation to show how users fared in completing a process, or to contextualize user feedback from usability tests. In a usability report, you might show where the major stumbling blocks were for users. When planning a usability test, you can use a flow chart to identify the anticipated path through the web site.



**Figure 9.10** This flow chart has been supplemented with icons and callouts to show where users struggled most in completing a process.

In capturing user needs, personas can use flow charts to illustrate typical processes, outside the context of a particular system. These flow charts map user expectations, showing how target audiences are used to completing a task, or how they expect related tasks to work together.

### Flow charts and strategy documents

In typical processes, flow charts documenting a web site's user experience come after strategy documents that establish a foundation for design. You may reference the competitive analysis or the concept model in the construction of the flow, but it's unlikely you'll draw any direct relationships between them.

On the other hand, flow charts can be useful in competitive analyses to show how different sites represent the same process. As entrenched as online shopping carts and checkout processes are, the user experience is slightly different on every commercial site. Lining up a series of flows that show the same task implemented in different ways can be a powerful way to draw attention to what your competitors do well.

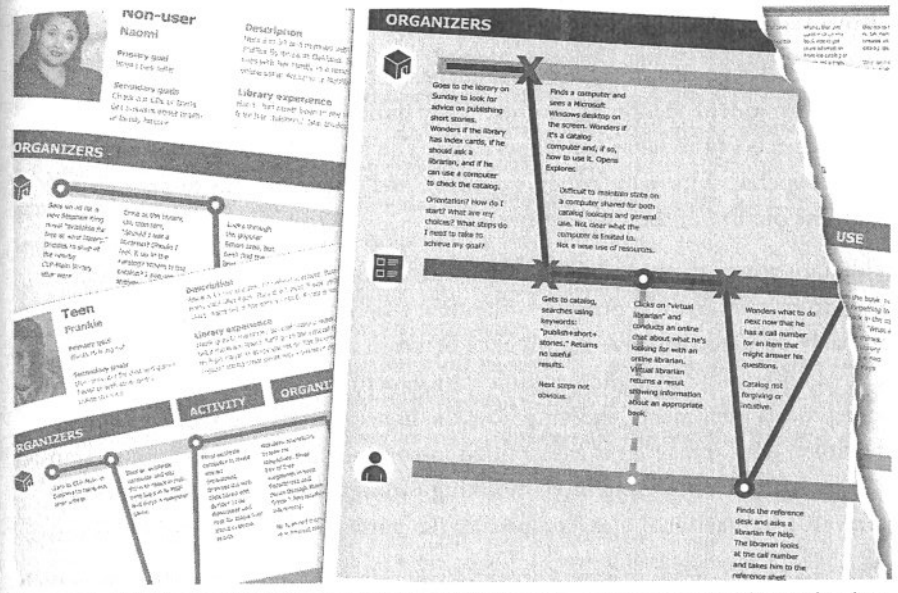
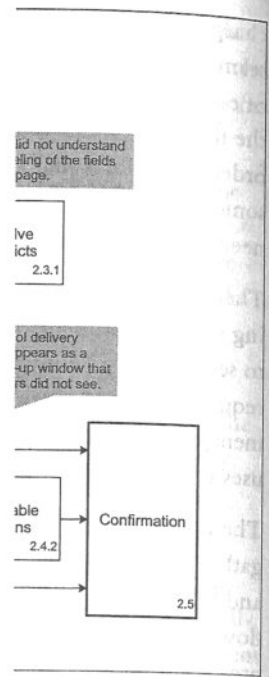


Figure 9.11 MAYA Design, in their work for the Carnegie Library system, used a stylized flow chart to show how users attempted to complete a task and where the library's user experience was "broken." Image courtesy of MAYA Design, www.maya.com.

how where users struggled

ustrate typical pro-  
ow charts map user  
ompleting a task, or

user experience come  
sign. You may refer-  
construction of the  
between them.

ve analyses to show  
red as online shop-  
is slightly different  
show the same task  
law attention to what

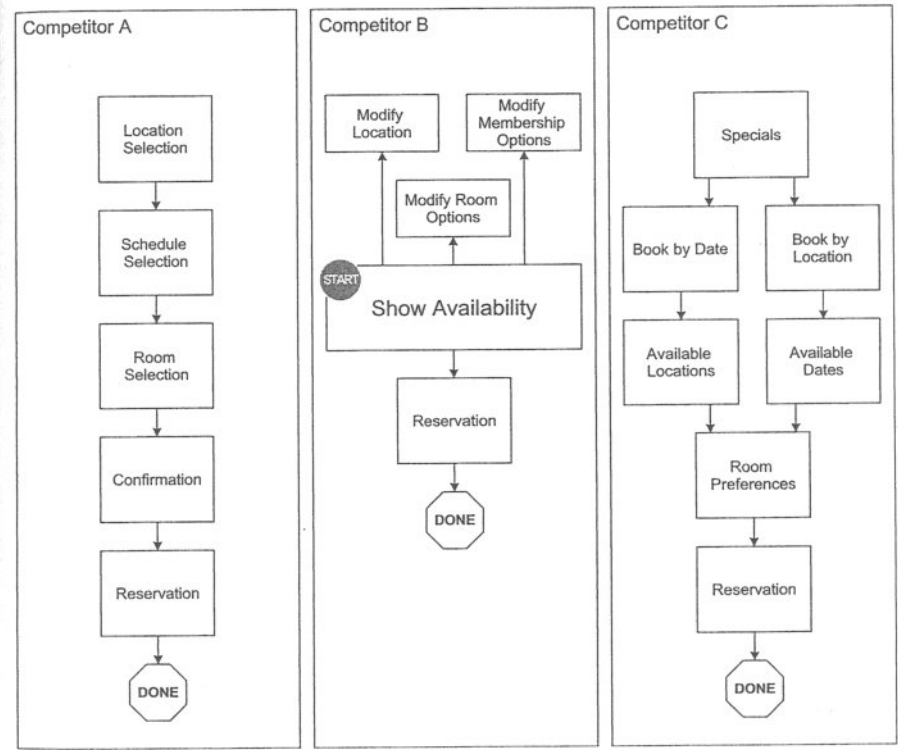


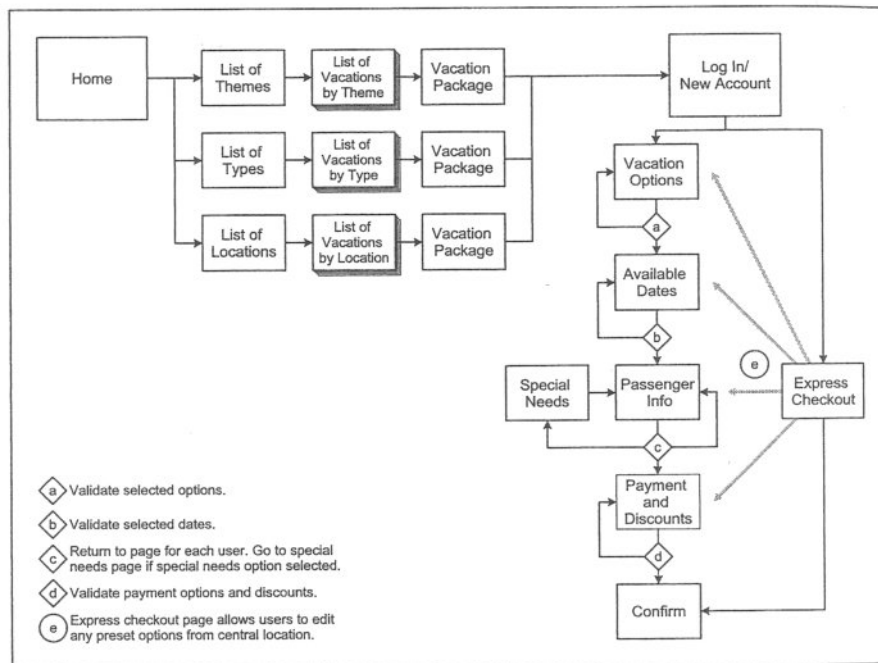
Figure 9.12 These three flows all illustrate the same task—booking a hotel room—but show how it has been implemented differently on different sites.

## Flow charts and other design documents

Successful design documentation cooperates. That is, no design document should live independently of another. They build on each other and show different aspects of the same user experience. The flow chart is no different. It must cooperate with wireframes, site maps, and screen designs to provide a complete picture of the user experience.

Although you may opt to treat the structure of the site (the site map) differently from transactional processes (the flow chart), keep in mind that the user probably does not perceive a difference between them. To better illustrate the user experience, you might combine your structural/navigation documents with your process documents, showing how the user experience flows from one to the other. This transition is exemplified in commercial sites, which must transition users from a browsing mode (looking through the shop to find a desired item) to a transactional one (completing the purchase of the item).

Flow charts also provide good context for wireframes and screen designs, which show a slice of the overall user experience. Integrating a flow chart into a wireframe can help stakeholders recognize where they are in the process. (This is discussed thoroughly in Chapter 11.)



**Figure 9.13** This complex site map/flow chart combination for a travel site shows how users move from browsing vacations to booking the trip.

If you ha  
combine  
arrows, y  
is very c  
to a fina  
they mig  
understa

The jur  
and exp  
this up  
Of cour  
cally or  
it can b

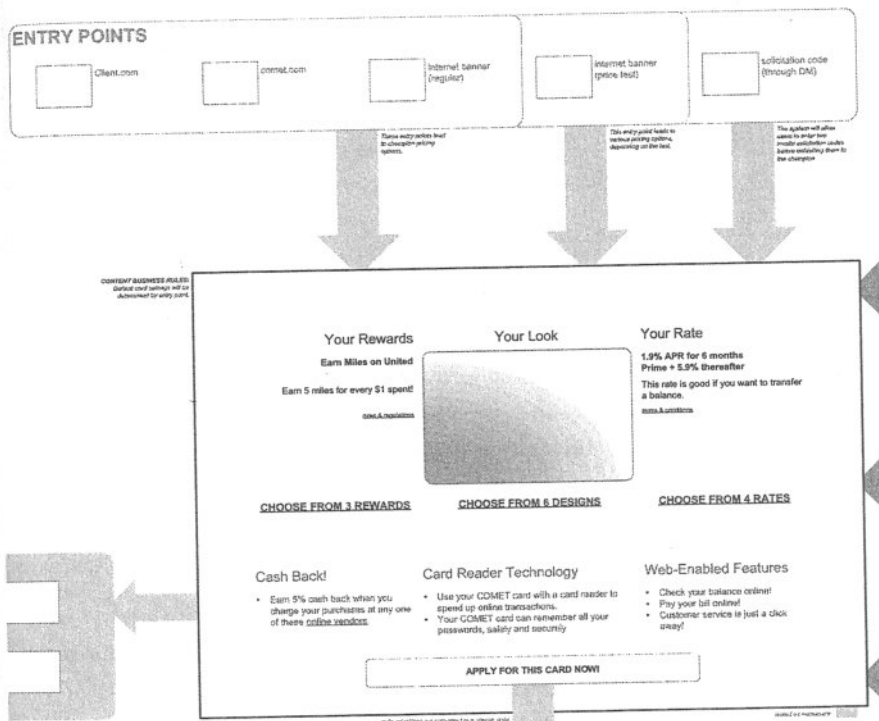


**Figure 9**  
opportu  
connect  
column



If you have access to a large-format printer, you can create a document that combines wireframes and flow charts. Instead of connecting rectangles with arrows, you connect entire wireframes with arrows. This kind of document is very cool because it reduces the abstraction, creating something that's closer to a final product. People reading this hybrid flow won't lose the context that they might when looking at individual wireframes, nor will they have trouble understanding a flow chart that lacks user interface details.

The jumbo format gives you more room to include additional documentation and explain the business rules that dictate what users see. Putting a poster like this up on a wall is great for collaboration because it invites people to annotate. Of course, the size comes with several disadvantages: It's not as portable, physically or electronically; it's difficult to make without a large-format printer; and it can be difficult to maintain.



**Figure 9.14** Freed from standard-sized paper, the wireframe/flow chart hybrid offers design teams countless opportunities to document every aspect of the user experience. This excerpt shows a couple of wireframes connected by arrows, elaborated business rules, and detailed behavior notes that aren't confined to a narrow column on one side of the page.

Weigh the disadvantages and decide if you can deal with them. You may need to create two documents from the same content—a large poster and a deck of wireframes. The poster may serve as a means for discussing the user experience, but the wireframe deck is the final word, and is the repository of the most up-to-date decisions.

### *The Many Levels of Process*

In the early days of the web, sites were just repositories of information, where the structure of navigation was paramount. As the web evolves, permitting different types of interactions, designers and developers will need a means for planning and documenting these complex user experiences. Flow charts will become increasingly important as web sites become more interactive.

But there's another aspect of the flow chart that makes it important to the design process. Internet technologies are playing an increasingly strategic role in business, becoming an essential plank in the customer communications and internal business operations platforms. Process documentation is crucial because the web changes the nature of interactions with customers and between colleagues. The flow chart allows us to get our heads around not just how people interact with the web site, but operations as a whole.

This isn't anything new—management consultants have been using flow charts for decades to describe the internal operations of an organization. But with the web comes an unprecedented level of interaction between people and technology. The web does more than allow us to conduct business differently, it has changed the face of what business is. As work becomes increasingly focused on information, the kinds of interactions organizations have with their customers and the kinds of collaboration that takes place between workers change dramatically. Workers are no longer cogs in a machine with specific tasks to accomplish, and customers are no longer held at arm's length with minimal impact on the organization and the work.

The web, as a business tool, forces us to think about how information moves in and out of the organization. The flow chart illustrates this experience.

The web, as a catalyst for change, forces us to rethink the roles of workers and customers, and their contributions to conducting business. In this case, the flow chart needs to operate at a higher level, showing how organizations must take advantage of and participate in these kinds of interactions. To accommodate this change, our notion of a flow chart must also change. Flow and process will

never g  
steps n:



Figure 1  
like this  
person

never go away, but our assumption that flows are linear and made of discrete steps must change to recognize the evolution of business.



**Figure 9.15** The new face of work. As the web changes how we do business, we're going to see more scenes like this one, where process isn't linear and collaboration is more than just handing off documents from one person to the next. Photo © Bethany Del Lima.