Scenarios

A **scenario** is a plausible description of the future based on a coherent set of assumptions. Sce-narios of one kind or another have long been used in many fields, such as military tactics or urban planning. In these arenas, a scenario is some combination of circumstances that *might* hap-pen; by imagining a range of possible scenarios, leaders can ensure that they are prepared for any eventuality. Scenarios are among the most power-ful tools in product and service design, with uses ranging from developing requirements to ensuring that a design accounts for the full range of pos-sible interactions.

A Goal-Directed scenario is a textual description of a persona's interaction with the future product or service. Each scenario begins with a specific situation, then describes the interaction between persona and system from the beginning of a task or session through its completion. A scenario describes the actions of the persona and any system behaviors or actions evident to that persona, but does not describe any system behavior that's invisible. Along the way, a good scenario explains the persona's motivations for particular behavior and indicates what persona goals the system's behavior achieves. In this way, scenarios are es-sential to realizing the value of personas: perso-nas without scenarios would be like characters with no plot.

The early scenarios used in requirements defini-tion, called **context scenarios**[1], are high level and optimistic, focusing on ideal system behavior in situations that *will* happen. Context scenarios do not include specific solutions. Later in the design process, increasingly detailed scenarios account for each constraint and for less common situations that *might* happen. These other scenarios—called key path, validation, and communication scenarios—are covered in later chapters.

Why use scenarios?

Context scenarios are the first activity in which you'll begin to imagine what the product or service could be like. If you're like most people, you learned to put away your imagination as you grew up, so these days you probably see an empty cardboard box as a recycling chore. A child, on the other hand, can see that box as a castle, a spaceship, or a bus. A favorite teddy bear might be anything from a hospital patient to a superhero. Each idea is inevitably accompanied by a story. By making use of these childhood tendencies, storytelling can help us imagination-impaired grown-ups remember how to see the possibilities in any situation. Like personas, scenarios can also help you evaluate whether your proposed solutions make sense.

Some people object to scenarios on the basis that they are invented. In truth, scenarios are more extrapolation than invention; they rely on our human understanding of a particular sort of person. You know a persona would behave in a particular way because you know what makes him tick. It's just like buying a gift for someone; the better you know that person, the more likely you are to find something they'll love. It may not be science, but it works all the same.

The third important virtue of scenarios is that they involve a *sequence* of events. Many talented but inexperienced interaction designers are inclined to draw screens and controls based on conceptual relationships rather than flow. Interactive products change state over time, though, so a sense of what comes first and what comes next is as essential here as it is to filmmaking.

Finally, scenarios provide a concrete way to think about human behavior and needs and their implications for system behavior. It's possible to argue forever about abstract concepts, but getting to specific examples can cut the argument short. When someone says, "The mobile phone should accept input from the television remote control," you can respond by asking, "Can you describe a scenario in which that would be useful to one of our personas?"

How Goal-Directed scenarios differ from similar tools

Scenarios and their close cousins, use cases, have seen increasingly widespread use in the last couple of decades. This is a good thing, since thinking about human behavior in sequence facilitates clear thinking about product behavior. Unfortunately, it also means that people are using similar terminology to mean slightly different things, so it's important to clarify how Goal-Directed scenarios differ from similar tools.

The term "scenario" is used in a variety of ways. Some people describe existing workflows as scenarios. There is disagreement about how detailed scenarios should be, whether they include system behavior that isn't visible to users, and how they should be documented. It's also a common misconception that scenarios, use cases, and user stories are interchangeable.

As originally defined by Ivar Jacobson and others, **use cases** describe the interaction between an "actor" and a system. An **actor** may be a person or another system, such as a bank database providing information to an automatic teller machine. Use cases are generally described in the form of diagrams, often using specific conventions such as those outlined in UML (Unified Modeling Language). "Goals" in use cases are typically at a lower level than persona goals; some may be end goals, but most are tasks. Use cases do not incorporate how users feel about the interaction or

[1] Although used somewhat differently in Goal-Directed Design, the term is borrowed from John Carroll (*Scenario-based design: Envisioning work and technology in system development*; John Wiley, 1995).

why a particular behavior is good. They also don't employ personas; even a human actor is likely to be defined as a role.

Agile programming methods such as Scrum advocate another tool called a **user story**. On the surface, this sounds similar to a scenario, but user stories are much more like requirements than scenarios. They're typically no more than a couple of sentences. For example, a user story titled "Login" might say, "When a user logs in, the application asks for a user name and password. It offers the option to remember both." User stories don't describe a user's entire flow from start to finish—a scenario, in contrast, would describe login as something that happens along the way to accomplishing an end goal. They don't use personas. Like use cases, they also don't focus on how users think and feel.

A SCENARIO DESCRIBES THE FUTURE, NOT THE PRESENT

Correctly or not, many people documenting how users interact with existing tools refer to such descriptions as workflows, scenarios, or use cases. For the sake of clarity, in the Goal-Directed lexicon, scenarios are always focused on use of the future product or system. Current behavior is described in the personas and other models.

A SCENARIO DESCRIBES A PERSONA'S POINT OF VIEW

A scenario represents the world as *users* will see it, deliberately excluding any behind-the-scenes work the system does. Describing the fact that the ATM checks with the central bank computer to determine how much cash a user can withdraw, for example, is problematic for a couple of reasons. First, it pulls the focus of the story away from visible behavior and into implementation, which is not within your purview as a designer. More important, it forces the people generating and reading the scenario to think in system terms instead of human terms; as discussed in Chapter 11 with respect to personas, thinking in human terms engages a different part of our brains. The fact that the ATM has to talk to the bank is a requirement—an implication of the story—but not part of the story itself. Just as you want people to agree that the personas are correct before you introduce controversial topics like what the personas require, you want people to see the possibilities in the story before they worry about what it takes to make the story come true.

Emotions and motivations are also part of a persona's point of view. If you are describing a parent uploading precious baby photos from a digital camera, it's important to understand that person's anxiety about making sure the photos are copied before deleting them from the camera. Describing this sort of thing helps ensure that the design team is using persona goals and characteristics to drive the system's behavior, while also ensuring that people who read the scenario understand why certain system behavior is important.

A SCENARIO IS A STORY WITH A BEGINNING AND AN END

Since they're stories, scenarios are generally expressed in narrative form (though it's not necessary to document every tiny scenario used in the design process). Users don't think, "I'm going to use the print function now," so scenarios don't describe the product in those terms. Rather, they describe an entire session or typical task based on what the persona would see as the task's beginning and end; this could involve a single function or several dozen, and might cover a few minutes or an entire day. A scenario might even describe interrupted usage, such as a task that someone starts in the morning and has to pick up again after half-a-dozen interruptions.

Crafting effective context scenarios

Context scenarios are the most ambitious and least detailed scenarios in the design process. A context scenario begins with some kind of triggering event, such as the receipt of a purchase requisition, the need to upload images because a camera's memory card is full, or the need to withdraw cash for the bus. From there, it describes the sequence of information exchange between persona and system, decisions made or actions taken by the persona, and the results the persona sees. Here's an example of a context scenario for a mobile smart phone and personal organizer:

A context scenario begins with a triggering event and goes on to describe the sequence of information exchange, actions, and results.

> After a long meeting, Anne pulls out her Personal Assistant to note a couple of items she needs to follow up on, confirm the location of her next meeting, and see if anything important has come up in the last couple of hours. When she turns on the screen, the PA shows her the subject and location of her next meeting, which is in 25 minutes. There's also an indication that she has three messages marked urgent (including one from her boss), one message from a client whose messages she's told the PA are top priority, and a dozen others that can probably wait.
>
> After noting her to-do items before she forgets them, Anne selects the urgent message from her boss, which is a voicemail, and listens to it as she walks to the parking garage. His question about a recent contract is time sensitive, so she selects the option to call him back. As soon as she's done answering his question, she looks to see who the other urgent messages are from and decides to ignore them for now. She selects the message from the important client. It's an e-mail, but she wants it read to her because she's fumbling to find her car keys. Deciding it doesn't need an immediate response, she tells the PA to remind her to follow up later today; she juggles so many things in a day that she needs help keeping track of the details.
>
> Getting into the car, she sees that she has 15 minutes left to get to her next meeting. It's potentially a large account, so she's anxious to arrive on time. She asks the PA for the fastest route from her current location. The PA shows her the best option based on current traffic conditions. Pulling out of the garage, she tells the PA to give her audio directions so she can keep her eyes on the road.

It's essential to take a solution-agnostic approach in context scenarios to avoid unconsciously eliminating options.

> Arriving at her destination right on time, Anne reviews the meeting participants so she can greet them by name; the personal touch is everything in sales. When she's escorted into the conference room, she sets her PA on the table in case she needs it. She knows the device won't interrupt her meeting, even by vibrating, unless someone tells her voicemail it's an emergency. Anne realizes a few minutes later that she needs some information from her desktop PC back at the office. She uses the PA to access the spreadsheet she needs.
>
> After wrapping up another successful meeting, Anne checks her PA again. With an hour until her next stop, she asks it to show her the way to the nearest café so she can grab a bite. The PA shows her a couple of options. Anne chooses the nearest and walks there using the PA's directions. She has a sandwich and a cup of tea as she uses the PA to check out the latest news headlines. Knowing it will take her 20 minutes to get to her next appointment, the PA interrupts Anne's reading when she has 30 minutes to go.
>
> After an afternoon of meetings, Anne checks for messages from her family. She sees an e-mail from her husband, Ted. She checks it in case there's something he wants from the grocery store. He wants her to pick up a pizza, but didn't specify what kind. She chooses the option that lets her respond to the message with a phone call. After a quick conversation, she hangs up and adds a veggie supreme to the grocery list on her PA. One more stop and she can go home.

You can see that the scenario is high level; it specifies a few key bits of information, but doesn't go into detail about whether each message has a subject line or whether driving directions are shown alongside a map. You can also see that the scenario doesn't specify mechanisms; instead, it treats the interaction between Anne and her Personal Assistant as a conversation. Phrases like "chooses an option" or "shows an indication that she has messages" imply what the functionality is, but don't specify a particular way to accomplish it. This solution-agnostic approach is essential in context scenarios, because jumping to solutions too early means eliminating options, often without even realizing you're doing that. In the PA example, it's tempting to specify that she uses a touch screen or a thumb keyboard. However, both of these choices have design and cost implications, so it's better not to make that assumption yet.

The following sections describe how to go about creating your context scenarios.

STEP 1: IDENTIFY WHAT CONTEXT SCENARIOS YOU NEED

Nearly every product or service involves multiple context scenarios.

You will almost always need at least a couple of context scenarios for any product or service. Complex authoring tools or multi-role enterprise products may involve a dozen context scenarios. It's rare to have much more than that unless you're designing something truly immense, such as an entire ERP system that covers nearly every function in a business; even so, you'd still be unlikely to have more than a double handful of scenarios per primary persona's interface. These small numbers may be shocking to people accustomed to documenting every possible use case. Not to worry: The design will eventually address even the more obscure possibilities. The idea with context scenarios is to focus on the big issues first and get agreement on them so you can start designing, rather than getting bogged down in analysis paralysis by trying to cover everything.

Your context scenarios should include a typical example of each major activity your personas are likely to engage in. Activities that tend to occur together should be combined in a single scenario. Scenarios for multiple small tasks, such as processing invoices, should cover several examples to illustrate how the persona would want to manage the flow from one invoice to another. Activities that occur in separate sessions or are not at all related are best addressed in distinct scenarios. Infrequent activities, such as configuration, are usually straightforward enough to address later. The exception is when the out-of-box experience is crucial, such as for a consumer gadget, or when painful configuration is one of the critical problems with an existing system.

Start by listing the critical activities of your primary personas. Determine whether any of those key tasks tend to occur together; if they do, assume that you should combine those in a single scenario. Once you have a list based on your primary, consider what additional activities you'll need to cover for any secondary personas. This includes not only additional tasks not performed by the primary, but also tasks that your secondary personas approach differently. For example, if two surgeon personas start the same surgery at different points, you would need to describe a context scenario for each. There's usually at least one scenario per secondary persona, even if it's a simple variation on an activity performed by the primary persona. Supplemental, negative, served, and customer personas do not appear in context scenarios (except to the extent that served personas are being helped by a primary or secondary). Table 12.2 outlines some example lists of context scenarios that should be outlined for various products.

Table 12.2. Example lists of context scenarios.

Product	Persona	Scenarios
E-mail system	A system administrator with simple needs (primary administrator)	— Set up the system — Add an account — Change settings — Delete an account — Upgrade the system
	A system administrator who makes complex connections to other systems (secondary administrator)	— Set up the system
	Someone who uses e-mail in a single location (primary end user)	— First use at the beginning of the day — Use throughout the day
	A mobile e-mail user (secondary end user)	— Remote use
Consumer digital camera	A family photographer of average skill (primary)	— Out-of-box experience — Taking photos at an event — Taking photos here and there — Uploading photos
	A hobbyist photographer with high standards who takes a lot of photos (secondary)	— Photo shoot — Uploading photos
Camera company Web site	An uninformed point-and-shoot buyer who doesn't want a lot of detail (primary for point and shoot content)	— Find a point-and-shoot camera that meets some basic needs and learn where to buy it
	An uninformed SLR buyer who needs help making a good choice (primary for SLR content)	— Find the right SLR and accessories and learn where to buy it
	A knowledgeable SLR buyer who wants to know a lot of technical detail (secondary for SLR content)	— Find the right SLR and accessories
	A current owner (primary for support content)	— Find a lens or accessory — Get help with a problem
	A camera dealer (primary for dealer content)	— Learn about the latest models — Set up a dealer account — Place an order — Handle a problem with an order
	A job seeker (primary for career content)	— Learn what's available and apply
Inbound call center software	An experienced call center agent (primary agent)	— Handle basic calls — Escalate a call
	An inexperienced agent (secondary agent)	— Handle basic calls — Escalate a call
	An escalation agent (secondary agent)	— Handle calls
	A supervisor in a small call center (secondary manager)	— Monitor call flow — Optimize operations — Coach an agent
	A manager of a large call center (primary manager)	— Monitor call flow across multiple units — Optimize operations across multiple units
Complex purchasing application	A person requesting a purchase (primary requester)	— Request a purchase
	A specialized purchasing agent for a manufacturing supply chain (primary purchasing agent)	— Process requests — Follow up on orders
	A specialized purchasing agent for miscellaneous corporate needs (secondary purchasing agent)	— Process requests — Follow up on orders
	A specialized goods-receipt clerk (primary for goods receipt)	— Process received shipments
	A specialized accounts-payable clerk (primary for accounts payable)	— Pay invoices, including a problem invoice
	An office manager (secondary for all three)	— Process purchase requests — Process received shipments — Pay invoices

Product	Persona	Scenarios
Family calendaring system	An at-home parent who manages a calendar (primary)	– Reviewing everyone's commitments and planning the day – Entering an upcoming event for a child – Finding a time when the whole family can do something with friends
	A working parent who manages the calendar (secondary)	– Accessing the calendar remotely
	A twelve-year-old (secondary)	– Adding an event
Device used to deliver intravenous medications in a hospital	Nurse in a general ward (primary)	– Administer a medication (simple case) – Administer multiple medications at a time (complex case) – Adjust dosage – Respond to a problem
	Nurse in a neonatal intensive care unit (secondary)	– Administer a medication
	Nurse in an oncology unit (secondary)	– Administer a medication
	Anesthesiologist in an operating room (secondary)	– Administer a medication while constantly monitoring vitals
	Nursing aide (secondary)	– Monitor – Respond to a problem
	Person setting up medication lists and safety parameters (primary administrator)	– System setup
	Person cleaning and servicing the device (primary maintenance)	– Clean the device – Replace parts
Clothing store targeting women	A brand-focused shopper (primary)	– Browse the store, try some things on, buy some – Look for a specific item – Return an item – Order an item that's not in stock
	A price-conscious shopper who can only afford sale items (secondary)	– Find items on sale
	A man shopping for a gift (secondary)	– Get help finding the right gift

STEP 2. DEVELOP EACH STORY

For each situation you've listed, the next step is to tell the story of the persona using the product or service. This requires a combination of imagination, empathy with the persona, and an understanding of some fundamental design principles; although you are not yet drawing solutions, you are beginning to make design decisions by describing the overall flow.

Answer the right questions

A context scenario answers most of the basic journalistic questions: who, what, where, when, why, and how. *Who*, of course, is the persona, but might also include other people with whom the persona interacts to accomplish a task. *What* is the data exchanged among the persona, other people, and the system, and the actions the persona takes as a result. *Where* is the setting in which the product or service will be used, such as in a typical office, in a meeting room, at home, or in someone's car. It could also include a description of a virtual workspace, such as a workstation or task queue shared by multiple people. *When* is an indication of whether this activity is common, how long it lasts, and whether it's interrupted. *Why* is the persona's reason for performing the task and his motivation for various behaviors along the way.

How is the problematic question; the scenario should outline the *process* the persona follows to complete the task, but not specify the particular *tools* used during that process. In other words, we need to know that she somehow gets the nail into the piece of wood, but don't care if it involves a hammer, a wrench, or the heel of her shoe.

Use the right level of detail

The trickiest part about context scenarios is judging how much detail is too much. Here's an example of a description that's not detailed enough:

> After a long meeting, Anne pulls out her Personal Assistant to note a couple of items she needs to follow up on, confirm the location of her next meeting, and see if anything important has come up in the last couple of hours. She checks her messages and responds to one from her boss but ignores the others. Getting into the car, she asks the PA for the fastest route to her destination.

It's difficult to tell from this description what the implications of the design will be. In the good example provided earlier, it was evident that voicemail and e-mail were both listed in one place and that the PA was aware of her current location as well as traffic conditions. Both involve considerable business and technical issues, so it's best not to gloss over them. Also, it's hard to see how Anne makes a decision about what message to respond to. Overall, you begin to get a sense of bare bones functionality, but not of how the system should behave.

> Context scenarios should be compelling; they need to engage the imagination and help people see possibilities, not bog them down in detail.

Too much detail can be equally problematic. Here's an example:

> When Anne turns on the screen, the PA shows her the subject and location of her next meeting, which is in 25 minutes. There's also a list of messages that includes information about the sender, time, and subject; at the top are three marked urgent (including one from her boss). Under those is one message from a client whose messages she's specified as top priority in her address book, and a dozen others that can probably wait.
>
> Anne selects the urgent message from her boss, which is a two-minute voicemail, and listens to it as she walks to the parking garage. His question about a recent contract is time sensitive, so she selects the option to call him back; she could also send voicemail, send e-mail, delete his message, or flag it to follow up on later. As soon as she's done answering his question, she looks to see who the other urgent messages are from and decides to ignore them for now. She selects the message from the important client. It's an e-mail, but she wants it read to her because she's fumbling to find her car keys. She selects the "read aloud" button. Deciding it doesn't need an immediate response, she adds it to her to-do list for later follow-up.

The first thing you probably noticed is that this version is tedious to read. Context scenarios should be compelling; they need to engage the imagination and help people see possibilities, not bog them down in detail. Some of the information is irrelevant at this point; it may eventually be important to know all of the options available at the end of the message, but this is a trivial issue compared to unified messaging and traffic-aware driving directions. You might also see that at this level of detail, it's almost impossible not to delve into solutions; the text is beginning to describe screen layout, navigation, and buttons.

As with most aspects of the method, using the context scenarios in subsequent steps will help you create better ones the next time. However, you'll avoid the most common mistakes if you keep the following tips in mind.

Start with an optimistic mind-set

Looking ahead can sometimes change how stakeholders think about timelines, so think of the long term even if you're working on a near-term release or constrained redesign. If you're working on a product that will launch in the next year, consider what another year or two of effort might make possible. For a product three years out, think about what four or five years might accomplish. In other words, you should always be somewhat more ambitious than you think your timeline allows, but not so much more ambitious that you will lose credibility and frustrate other product team members. The work isn't wasted in any case, because you can use it again when the next version of the product ships or the service is expanded.

One way to ensure that you're thinking optimistically is to pretend that the thing you're designing is either magic or human. This gives you permission to forget about technology and constraints. You can see the results of this kind of thinking in the Personal Assistant example scenario described earlier. Take this sentence, for instance:

> There's also an indication that she has three messages marked urgent (including one from her boss), one message from a client whose messages she's told the PA are top priority, and a dozen others that can probably wait.

A helpful human assistant would prioritize messages so time-sensitive or important issues get dealt with sooner. Having senders mark messages as urgent is insufficient because the recipient may have different ideas about what is and isn't time-critical. It would take a lot for an electronic assistant to prioritize messages based on content, but it wouldn't be terribly hard for a system to give special treatment to messages from certain senders.

Exercise

What would the following things do if they were magical or replaced by a helpful human?

- Your e-mail application
- An automatic teller
- An airline reservation system
- A clothing Web site

Stay true to the personas

Typically, one team member sits at a keyboard (with the document projected on the wall) or stands at a whiteboard to capture and facilitate. Another team member drives the story forward. Any team member not proposing the scenario elicits more detail, throws in an occasional idea, and watches for constrained thinking or excessive specificity. Any team member might play either role, but most often the IxD generator or ID is inclined toward ideation, while the IxD synthesizer facilitates and acts as a backstop.

Team members also share a responsibility to challenge anything that's inconsistent with the persona's goals, attitudes, or behavior. Here's an example of how such a conversation might go:

> **Designer A:** Anne orders a sandwich and a cup of tea, then uses the Personal Assistant to read her other messages while she eats.
>
> **Designer B:** Why do you think she reads her other messages during lunch?
>
> **Designer A:** It's her first chance to look over the non-urgent ones.

Designer B: Sure, but one of her goals is to maintain some balance between work and life. It seems like she'd want to take a break and not be all about work, even if it's just for a few minutes.

Designer A: Hmmm ... good point. She'd probably read the newspaper or a magazine. Maybe the PA should make that kind of thing available to her.

Before moving on from one scenario to the next, see if anything in the scenario causes problems for other relevant user personas (either primary or secondary) or for your served or customer personas. If so, consider how you might modify it without making the star of the scenario unhappy.

Apply important design principles

As you describe the behavior of the system, you're beginning to design even though you're not getting specific about solutions. The important design principles to consider at this point are essentially descriptions of good product manners,[2] such as these:

- Do no harm, whether that's actual injury or just wasting someone's time.
- Provide all the tools your persona uses at the same time in the same place.
- Don't ask for confirmation of actions; make them possible to undo instead.
- Don't interrupt users to report events they don't care about.
- Make errors impossible; don't offer choices that won't work.
- Don't ask users to remember things if you can avoid it.
- Remember and learn from user behavior.
- Make reasonable assumptions instead of forcing users to customize or make a lot of unnecessary choices.
- Don't confuse what users will always do with what they might occasionally do.

Teammates should also be on the lookout for bad design decisions implied by a scenario. Here's an example:

Designer A: Anne listens to the urgent message from her boss. She decides she needs to reply, so she opens up her address book to find his number and call him back.

Designer B: Why would we force her to go to an address book application? Shouldn't we anticipate that she might want to respond either via e-mail or voice and make the option available right there?

Designer A: Hmmm ... yeah. You're right.

Have someone review your scenarios

It's easy when you develop the scenarios to get nearsighted about them, so have someone review your scenarios after you've completed them. A team lead who is knowledgeable about good scenario construction and has participated in part of the research is the best person for this. You may also want to have a subject matter expert or the project owner look over the scenarios before you extract requirements from them and share them with other stakeholders.

[2] For good examples of this kind of principle, see *About face 3: The essentials of interaction design*, by Alan Cooper, Robert Reimann, and David Cronin. Wiley. 2007.

Exercise

Critique the following context scenario for Carla Ramirez, the car buyer persona described in Chapter 11. Where is there too much detail, and where too little? What premature design decisions do you see? Where are good design principles not being followed? Where is the scenario not true to the persona?

CARLA LOOKS FOR A NEW CAR

Carla's MINI Cooper is starting to experience the mechanical problems that come with age, so she thinks it's time to buy a new car. She's torn between the CarCo Alpha and that little MegaMotors car she can never remember the name of (it's all numbers and letters). When she sits down at her home computer and opens up the MegaMotors Web site, she instantly recognizes an image of the PC3. She decides to look at more detail to see if it's the right car for her.

Carla goes to the PC3 page, which has several menu items across the top, including a gallery, specs, reviews, features, and accessories. First, though, she's drawn to the little widget that lets her spin the car around and change its color. After a minute of noodling with that, she thinks it's time to get down to business, and looks for more information on the car's interior by clicking the "features" menu item. When she finds her way to the interior specs, she sees that the trunk has eight cubic feet of cargo space; when she rolls over this, she sees that it translates to about four bags of groceries or a small roll-aboard suitcase and carry-on bag. It gets 30 miles to a gallon on the highway, which isn't bad.

Her patience for specs exhausted, Carla moves on to the "build your PC3" feature. She picks red paint, a tan interior, and a sun roof. The tool tells her the tan interior is not available with a red exterior, so she goes back to the previous step and picks a black interior. She adds a stereo upgrade, leather seats, and alloy wheels. Ouch, it's a little expensive. She sees an option to compare the PC3 to competitor's cars, so she decides to see how a comparably equipped Alpha would be priced. It's not much less, so Carla decides to see where the nearest dealer is so she can take a test drive this weekend.

Review your scenarios with your team lead and subject matter experts before extracting requirements.

STEP 3. PREPARE TO COMMUNICATE YOUR SCENARIOS

Most of the work in scenario creation happens in team meetings. However, the rough outline still has to be turned into a compelling narrative. Any team member may take on this task depending on who has time for it, but it's generally the province of the IxD synthesizer because crafting each sentence by committee would be painful. However, each team member should take a look at the draft narrative and comment on it before it gets shared outside the design team.

It's almost always a good idea to share your context scenarios with stakeholders, since they will be the basis for many of your requirements. There are rare stakeholders who don't respond well to scenarios due to a discomfort with things that aren't concrete. Verbal descriptions can create anxiety in people who aren't accustomed to imagining things they can't see. You generally have three choices for handling this situation. One is to provide lots of disclaimers about what scenarios can and can't do and encourage the anxious stakeholders to sit tight for a week or two until you have sketches. This can be uncomfortable, but doesn't usually result in big problems. Another is to jump straight to requirements without directly communicating the scenarios. This can work, but it's harder to justify how you arrived at your conclusions.

The third option is to illustrate your scenarios to make them more concrete. High-level storyboards can work well with scenarios involving movement through physical space, such as the use of a mobile device with another system. This is seldom a good idea with software interfaces, because it's hard to do without representing navigational structures and other details you haven't considered yet. If you believe you'll need to illustrate your scenarios, try to delay the stakeholder meeting for discussing requirements until you've had at least a few days to start the design framework (see Chapters 14 and 16).

A good storyboard at this level shows only the highlights of the interaction and deliberately glosses over (or crops out) anything you can't be specific about yet. If you don't have a good illustrator on your team, you can take photos of your teammates in the right poses, then use an application like Adobe Illustrator or Photoshop to draw a silhouette on top of the image. The one in Figure 12.3, for example, shows the context and the high points of Anne's "conversation" with her Personal Assistant, but it hand-waves the form factor and screen contents.

Exercise

Make a list of the context scenarios you need for the RoomFinder or LocalGuide, considering each of your personas. If you haven't created those personas, use Katie Bennett, the photographer persona from Chapter 11. Develop and document at least two context scenarios.

Extracting requirements from scenarios

Finally, you need to extract requirements from your scenarios. Even if it feels a bit repetitious, most stakeholders have an easier time digesting requirements that are called out in a list rather than hidden in a narrative. One effective way to do this is to put the scenario in a table and list the relevant requirements alongside it. You can then duplicate those requirements in a centralized list, if you have one. See Table 12.3 for an example.

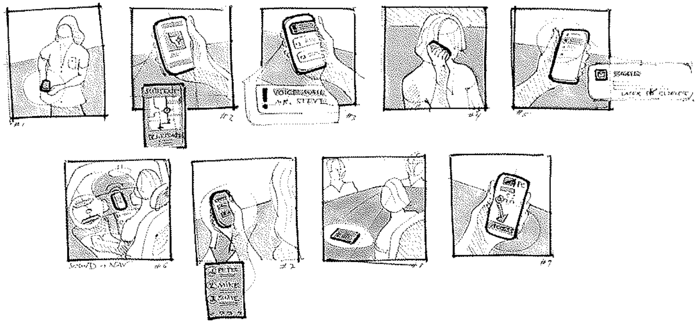

Figure 12.3. An example storyboard for Anne's Personal Assistant scenario.

Table 12.3. **Example requirements from a context scenario.**

Scenario text	Requirements
After a long meeting, Anne pulls out her Personal Assistant to note a couple of items she needs to follow up on, confirm the location of her next meeting, and see if anything important has come up in the last couple of hours.	– Ability to enter text – Ability to track appointments – Ability to see a list of messages – Portable form factor
When she turns on the screen, the PA shows her the subject and location of her next meeting, which is in 25 minutes.	– Ability to turn off the screen without the turning off the device – Ability to count down to the next event
There's also an indication that she has three messages marked urgent (including one from her boss), one message from a client whose messages she's told the PA are top priority, and a dozen others that can probably wait.	– Ability to see both e-mail and voice messages in a single place, along with next event – Ability to auto-prioritize some messages based on simple criteria specified by users, as well as based on urgency indicated by the sender

Continued

Scenario text	Requirements
After noting her to-do items before she forgets them, Anne selects the urgent message from her boss, which is a voicemail, and listens to it as she walks to the parking garage.	– Ability to enter and track tasks – Ability to select a message from a visual list – Ability to listen to voicemail
His question about a recent contract is time sensitive, so she selects the option to call him back.	– Ability to initiate various types of return communication directly from a message
As soon as she's done answering his question, she looks to see who sent the other urgent messages and decides to ignore them for now.	– Ability to return to what she was doing last
She selects the message from the important client. It's an e-mail, but she wants it read to her because she's fumbling to find her car keys.	– Ability to hear e-mail messages hands free
Deciding it doesn't need an immediate response, she tells the PA to remind her to follow up later today; she juggles so many things in a day that she needs help keeping track of the details.	– Ability to schedule action items or reminders from a message
Getting into the car, she sees that she has 15 minutes left to get to her next meeting. It's potentially a large account, so she's anxious to arrive on time.	– Ability to count down to the next event
She asks the PA for the fastest route from her current location.	– Ability to approximate current location closely enough to provide driving directions Ability to calculate fastest route from current location
The PA shows her the best option based on current traffic conditions.	– Ability to factor in current traffic conditions when calculating fastest route
Pulling out of the garage, she tells the PA to give her audio directions so she can keep her eyes on the road.	– Ability to get audio directions – Ability to provide appropriately timed driving directions
Arriving at her destination right on time, Anne reviews the meeting participants so she can greet them by name; the personal touch is everything in sales.	– Ability to review information about meeting participants

Scenario text	Requirements
When she's escorted into the conference room, she sets her PA on the table in case she needs it. She knows the device won't interrupt her meeting, even by vibrating, unless someone tells her voicemail it's an emergency.	– Ability to select parameters for interruptions and to apply them automatically during scheduled meetings
Anne realizes a few minutes later that she needs some information from her desktop PC back at the office. She uses the PA to access the spreadsheet she needs.	– Ability to connect to a remote computer with appropriate permissions – Ability to view common document formats
After wrapping up another successful meeting, Anne checks her PA again. With an hour until her next stop, she asks it to show her the way to the nearest café so she can grab a bite. The PA shows her a couple of options. Anne chooses the nearest and walks there using the PA's directions.	– Ability to count down to next event – Ability to locate common services such as food, fuel, etc.
She has a sandwich and a cup of tea as she uses the PA to check out the latest news headlines.	– Ability to get various publicly available content
Knowing it will take her 20 minutes to get to her next appointment, the PA interrupts Anne's reading when she has 30 minutes to go.	– Ability to get proactive reminders that are intelligent about accounting for travel time
After an afternoon of meetings, Anne checks for messages from her family. She sees an e-mail from her husband, Ted.	– Ability to view messages
She checks it in case there's something he wants at the grocery store. He wants her to pick up a pizza, but didn't specify what kind. She chooses the option that lets her respond to the message with a phone call.	– Ability to initiate a message response from one channel in any other channel, directly from the message
After a quick conversation, she hangs up and adds a veggie supreme to the grocery list on her PA. One more stop and she can go home.	– Ability to track various lists

As you can see, these requirements are framed in terms of what Anne needs to do. They're solution-agnostic and not terribly detailed. "Ability to approximate current location closely enough to provide driving directions," for example, doesn't specify that the device must know a location to within two meters, because it will take some experimentation to see just how much imprecision you can get away with. It also doesn't specify GPS technology, since it might be possible to triangulate an approximate location using cell phone towers or some other method.

You might also notice in the example that the ability to count down to the next event is listed more than once. You could certainly stop after you've listed a need once, but multiple listings can help people see that a certain capability has multiple uses and may show up in multiple scenarios. It also helps with making your requirements traceable.

Exercise

Extract requirements from your RoomFinder or LocalGuide scenarios, or from your rewritten scenario for Carla's car-buying experience.

Other Requirements from User Personas

The majority of your requirements will come from the scenarios, but persona goals, skills, environments, physical attributes, and mental models are all helpful in identifying additional needs. The list of requirements from each of these sources tends to be much shorter than the list generated from the scenarios.

Mental models

As discussed in the "Mental models" section of Chapter 7, a mental model describes how someone thinks about objects, relationships, and actions. A design that's antithetical to the personas' mental models will be difficult to learn and use, so mental models are often an important source of requirements. A photographer who thinks of her photos as belonging to a particular event, such as a holiday or vacation, will be frustrated if she can't group her photos based on events. On the other hand, someone who takes photos to create aesthetically pleasing images will be annoyed if he's forced to organize those into mutually exclusive albums. This example might also occur naturally in a scenario, but it's worth asking yourself whether you've missed any other requirements implied by the mental model.

Environments

Some requirements are suggested by the virtual or physical environment in which a product will be used, as well as the personas' expectations of what is normal for that environment. For example, hospital staff would be surprised if a medical device couldn't survive being scrubbed with soapy water and sprayed with bleach, whereas most reasonable people don't expect a mobile phone to work after it's gone through the washing machine. Typical environmental considerations include:

- Lighting conditions under which a screen should be comfortably viewable
- Viewing angle from which data on a screen should be readable
- Distance at which text or indicators must be readable
- Physical abuse and temperature variation a device should be able to survive
- Privacy needs for data viewed in public places
- Need to share task lists or other virtual spaces with other users
- Frequency of interruption
- Portability needs
- Whether input will be hampered by gloves
- Ambient noise level above which any sound should be audible
- Security needs, such as protecting settings from unauthorized changes
- Need to fit within a certain size display or storage space

Some of these requirements will evolve as the design and engineering progress, but the more specific you can be early on, the better.

Physical and cognitive characteristics

It's obvious that industrial designers need to worry about a persona's physical characteristics, such as the range of body sizes and physical capabilities a device should accommodate. However, this is also a consideration for interaction and visual design. People with poor vision will need larger text. Users with arthritis will need interactions that don't depend on a lot of manual dexterity or strength. Even users with perfect eyesight and a full range of motion need to avoid repetitive stress injuries.

The Measure of Man and Woman[3] and the *Handbook of Human Factors and Ergonomics*[4] are both useful references for understanding human physical and cognitive characteristics.

Consider what kind of support the personas need from the product or service, whether this support is about technology or about an area of expertise.

Skills and knowledge

Consider what kind of support the personas need from the product or service, whether this support is about technology or about an area of expertise. For example, an average person filling out his annual tax forms may need help understanding what deductions he can take, whereas a professional accountant would not. Someone looking up prescription drug information probably needs help spelling obscure medication names. Someone unpacking a new home computer might require help understanding how to set it up.

Goals

Goals are the other important source of persona-based requirements. Some of those requirements are intangible, such as helping someone feel confident in a financial institution. Katie Bennett, the photography persona from Chapter 11, wants to feel like a "real" photographer, which implies that the camera has to look like a professional camera and feel solid in her hands. (See the discussion of experience attributes later in this chapter.)

[3] Tilley, A.R. and Henry Dreyfuss Associates. *The measure of man and woman: Human factors in design*. Wiley, 2001.

[4] Salvendy, G., Editor. *Handbook of human factors and ergonomics*. Third edition. Wiley, 2006.