

Chapter 3

User Role Modeling

On many projects, stories are written as though there is only one type of user. All stories are written from the perspective of that user type. This simplification is a fallacy and can lead a team to miss stories for users who do not fit the general mold of the system's primary user type. The disciplines of usage-centered design (Constantine and Lockwood 1999) and interaction design (Cooper 1999) teach us the benefits of identifying user roles and personas prior to writing stories. In this chapter we will look at user roles, role modeling, user role maps, and personas and show how taking these initial steps leads to better stories and better software.

User Roles¹

Suppose we are building the BigMoneyJobs job posting and search site. This type of site will have many different types of users. When we talk about *user* stories, who is the user we're talking about? Are we talking about Ashish who has a job but always keeps an eye out for a better one? Are we talking about Laura, a new college graduate looking for her first professional job? Are we talking about Allan, who has decided he'll take any job that lets him move to Maui and windsurf every afternoon? Or are we talking about Scott, who doesn't hate his job but has realized it's time to move on? Perhaps we're talking about Kindra who was laid off six months ago and was looking for a great job but will now take anything in the northeastern United States.

Or should we think of the user as coming from one of the companies posting the jobs? Perhaps the user is Mario, who works in human resources and posts

1. Much of the discussion of user roles in this chapter is based on the work of Larry Constantine and Lucy Lockwood. Further information on user role modeling is available at their website at www.foruse.com or in *Software for Use* (1999).

new job openings. Perhaps the user is Delaney, who also works in human resources but is responsible for reviewing resumes. Or perhaps the user is Savannah, who works as an independent recruiter and is looking for both good jobs and good people.

Clearly we cannot write stories from a single perspective and have those stories reflect the experiences, backgrounds and goals of each of these users. Ashish, an accountant, may look at the site once a month just to keep his options open. Allan, a waiter, may want to create a filter to notify him any time any job on Maui gets posted but he won't be able to do that unless we make it easy. Kindra may spend hours each day looking for a job, broadening her search as time goes by. If Mario and Delaney work for a large company with many positions to fill, they may spend four or more hours a day on the site.

While each user comes to your software with a different background and with different goals, it is still possible to aggregate individual users and think of them in terms of *user roles*. A user role is a collection of defining attributes that characterize a population of users and their intended interactions with the system. So, we could look at the users in the preceding example and group them into roles as shown in Table 3.1 into roles this way.

Table 3.1 *One possible list of roles for the BigMoneyJobs project.*

Role	Who
Job Seeker	Scott
First Timer	Laura
Layoff Victim	Kindra
Geographic Searcher	Allan
Monitor	Ashish
Job Poster	Mario, Savannah
Resume Reader	Delaney, Savannah

Naturally, there will be some overlap between different user roles. The Job Seeker, First Timer, Layoff Victim, Geographic Searcher, and Monitor roles will all use the job search features of the site. They may use them in different ways and at different frequencies, but much of how they use the system will be similar. The Resume Reader and Job Poster roles will probably overlap as well since these roles are both pursuing the same goal of finding good candidates.

Table 3.1 does not show the only possible way to group users of BigMoneyJobs into roles. For example, we could choose to include roles like Part-Timer,

Full-Timer and Contractor. In the rest of this chapter we'll look at how to come up with a list of roles and how to refine that list so that it is useful.

Role Modeling Steps

We will use the following steps to identify and select a useful set of user roles:

- brainstorm an initial set of user roles
- organize the initial set
- consolidate roles
- refine the roles

Each of these steps is discussed in the following sections.

Brainstorming an Initial Set of User Roles

To identify user roles, the customer and as many of the developers as possible meet in a room with either a large table or a wall to which they can tape or pin cards. It's always ideal to include the full team for the user role modeling that initiates a project but it's not necessary. As long as a reasonable representation of the developers is present along with the customer, you can have a successful session.

Each participant grabs a stack of note cards from a pile placed in the middle of the table. (Even if you plan to store the user roles electronically you should start by writing them on cards.) Start with everyone writing role names on cards and then placing them on a table, or taping or pinning them to a wall.

When a new role card is placed, the author says the name of the new role and nothing more. Since this is a brainstorming session, there is no discussion of the cards or evaluation of the roles. Rather, each person writes as many cards as he or she can think of. There are no turns, you don't go around the table asking for new roles. Each participant just writes a card whenever she thinks of a new role.

While brainstorming roles, the room will be filled with sounds of pens scratching on cards and will be punctuated by someone occasionally placing a new card and reading the name of the role. Continue until progress stalls and participants are having a hard time thinking up new roles. At that point you may not have identified all of the roles but you're close enough. Rarely does this need to last longer than fifteen minutes.

A User Role Is One User

When brainstorming a project's roles, stick to identifying roles that represent a single user. For example, for the BigMoneyJobs project it may be tempting to write stories such as "A company can post a job opening." However, since a company as a whole cannot use the software, the story will be better if it refers to a role that represents an individual.

Organizing the Initial Set

Once the group has finished identify roles, it's time to organize them. To do this, cards are moved around on the table or wall so that their positions indicate the relationships between the roles. Overlapping roles are placed so that their cards overlap. If the roles overlap a little, overlap the cards a little. If the roles overlap entirely, overlap the cards entirely. An example is shown in Figure 3.1.

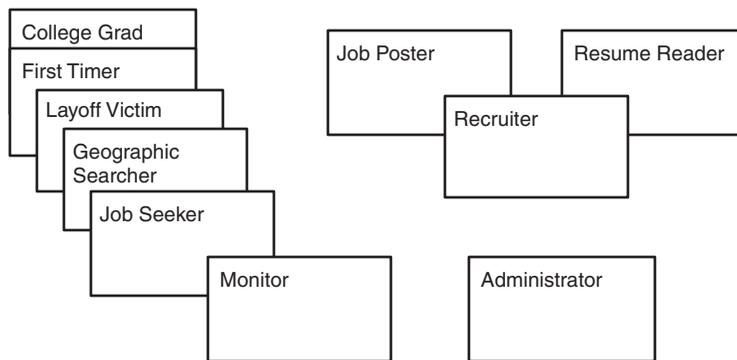


Figure 3.1 *Organizing the user role cards on a table.*

Figure 3.1 shows that the College Grad and First Timer, as those roles were intended by their card writers, overlap significantly. There's less but similar overlap among the other cards representing people who will use the site to search for jobs. The Monitor role card is shown with only a slight overlap because that role refers to someone who is relatively happy in her current job but likes to keep her eyes open.

To the right of the job-seeking roles in Figure 3.1 are the Job Poster, Recruiter, and Resume Reader role cards. The Recruiter role is shown overlapping both Job Poster and Resume Reader because a recruiter will both post ads

and read resumes. An Administrator role is shown also. This role represents users internal to BigMoneyJobs who will support the system.

System Roles

As much as you can, stick with user roles that define people, as opposed to other systems. If you think it will help, then identify an occasional non-human user role. However, the purpose of identifying user roles is to make sure that we think really hard about the users that we absolutely, positively must satisfy with the new system. We don't need user roles for every conceivable user of the system, but we need roles for the ones who can make or break the success of the project. Since other systems are rarely purchasers of our system, they can rarely make or break the success of our system. Naturally, there can be exceptions to this and if you feel that adding a non-human user role helps you think about your system, then add it.

Consolidating Roles

After the roles have been grouped, try to consolidate and condense the roles. Start with cards that are entirely overlapping. The authors of overlapping cards describe what they meant by those role names. After a brief discussion the group decides if the roles are equivalent. If equivalent, the roles can either be consolidated into a single role (perhaps taking its name from the two initial roles), or one of the initial role cards can be ripped up.

In Figure 3.1 the College Grad and First Timer roles are shown as heavily overlapping. The group decides to rip up the College Grad card since any stories for that user role would likely be identical to stories for a First Timer. Even though First Timer, Layoff Victim, Geographic Searcher and Job Seeker have significant overlap, the group decides that each represents a constituency that will be important to satisfy and the roles will have important but subtly different goals for how they use the BigMoneyJobs website.

When they look at the right side of Figure 3.1, the group decides that it is not worth distinguishing between a Job Poster and a Resume Reader. They decide that a Recruiter covers these two roles adequately and those cards are ripped up. However, the group decides that there are differences between an Internal Recruiter (working for a specific company) and an External Recruiter (matching candidates to jobs at any company). They create new cards for Internal Recruiter and External Recruiter, and consider these as specialized versions of the Recruiter role.

In addition to consolidating overlapping roles, the group should also rip up any role cards for roles that are unimportant to the success of the system. For example, the Monitor role card represents someone who is just keeping an eye on the job market. A Monitor may not switch jobs for three years. BigMoneyJobs can probably do quite well without paying attention to that user role. They decide they will be better off focusing on the roles that will be important to the success of the company, such as Job Seeker and the Recruiter roles.

After the team has consolidated the cards, they are arranged on the table or wall to show relationships between the roles. Figure 3.2 shows one of many possible layouts for the BigMoneyJobs role cards. Here a generic role, such as Job Seeker or Recruiter, is positioned above specialized versions of that role. Alternatively, cards can be stacked or positioned in any other way that the group desires in order to show whatever relationships they think are important.

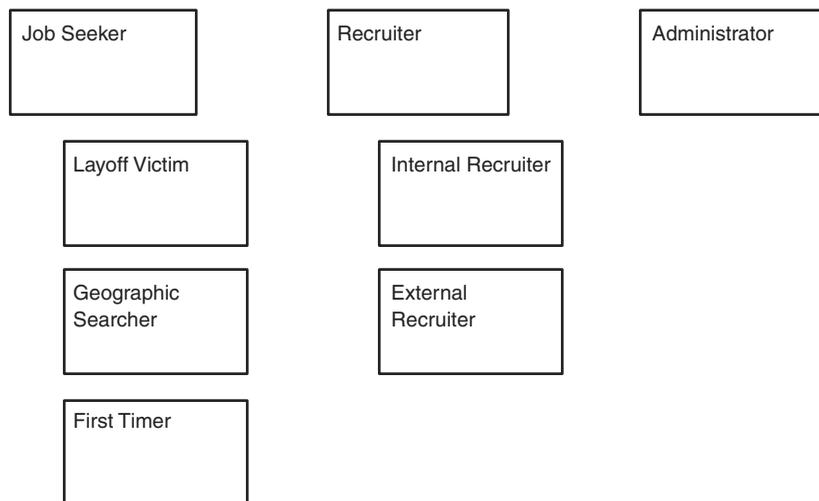


Figure 3.2 *The consolidated role cards.*

Refining the Roles

Once we've consolidated roles and have a basic understanding of how the roles relate to each other, it is possible to model those roles by defining attributes of each role. A role attribute is a fact or bit of useful information about the users who fulfill the role. Any information about the user roles that distinguishes one role from another may be used as a role attribute. Here are some attributes worth considering when preparing any role model:

- The frequency with which the user will use the software.
- The user's level of expertise with the domain.
- The user's general level of proficiency with computers and software.
- The user's level of proficiency with the software being developed.
- The user's general goal for using the software. Some users are after convenience, others favor a rich experience, and so on.

Beyond these standard attributes you should consider the software being built and see if there are any attributes that might be useful in describing its users. For instance, for the BigMoneyJobs website you may want to consider whether the user role will be looking for a part-time or full-time job.

As you identify interesting attributes for a role, write notes on the role card. When finished, you can hang the role cards in a common area used by the team so they can be used as reminders. A sample user role card is shown in Figure 3.3.

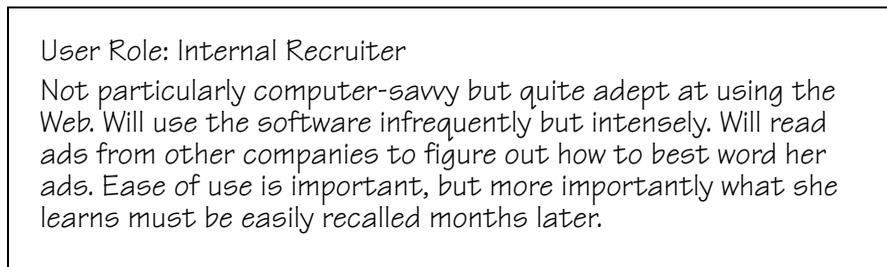


Figure 3.3 *A sample user role card.*

Two Additional Techniques

We could stop right now if we want to. By now a team might have spent an hour—almost certainly no more than that—and they will have put more thought into the users of their software than probably 99% of all software teams. Most teams should, in fact, stop at this point. However, there are two additional techniques that are worth pointing out because they may be helpful in thinking about users on some projects. Only use these techniques if you can anticipate a direct, tangible benefit to the project.

Personas

Identifying user roles is a great leap forward, but for some of the more important user roles, it might be worth going one step further and creating a *persona* for the role. A persona is an imaginary representation of a user role. Earlier in this chapter we met Mario who is responsible for posting new job openings for his company. Creating a persona requires more than just adding a name to a user role. A persona should be described sufficiently that everyone on the team feels like they know the persona. For example, Mario may be described as follows:

Mario works as a recruiter in the Personnel department of Speedy-Networks, a manufacturer of high-end networking components. He's worked for SpeedyNetworks six years. Mario has a flex-time arrangement and works from home every Friday. Mario is very strong with computers and considers himself a power user of just about all the products he uses. Mario's wife, Kim, is finishing her Ph.D. in chemistry at Stanford University. Because SpeedyNetworks has been growing almost consistently, Mario is always looking for good engineers.

If you choose to create personas for your project, be careful that enough market and demographic research has been done that the personas chosen truly represent the product's target audience.

This persona description gives us a good introduction to Mario. However, nothing speaks as loudly as a picture, so you should also find a picture of Mario and include that with the persona definition. You can get photographs all over the web or you can cut one from a magazine. A solid persona definition combined with a photograph will give everyone on the team a thorough introduction to the persona.

Most persona definitions are too long to fit on a note card, so I suggest you write them on a piece of paper and hang them in the team's common space. You do not need to write persona definitions for every user role. You may, however, think about writing a persona definition for one or two of the primary user roles. If the system you are building is such that it is absolutely vital that the product satisfy one or two user roles, then those user roles are candidates for expansion into personas.

Stories become much more expressive when put in terms of a user role or persona. After you have identified user roles and possibly a persona or two, you can begin to speak in terms of roles and personas instead of the more generic "the user." Rather than writing stories like "A user can restrict job searches to specific geographic regions" you can write "A Geographic Searcher can restrict

his job searches to a specific geographic region.” Hopefully writing a story this way reminds the team about Allan who is looking for any job on Maui. Writing some stories with user role or persona names does not mean that other roles cannot perform those stories; rather, it means that there is some benefit in thinking about a specific user role or persona when discussing or coding the story.

Extreme Characters

Djajadiningrat and co-authors (2000) have proposed a second technique you might want to think about: the use of extreme characters when considering the design of a new system. They describe an example of designing a Personal Digital Assistant (PDA) handheld computer. They advise that instead of designing solely for a typical sharp-dressed, BMW-driving management consultant, the system designers should consider users with exaggerated personalities. Specifically, the authors suggest designing the PDA for a drug dealer, the Pope, and a twenty-year-old woman who is juggling multiple boyfriends.

It is very possible that considering extreme characters will lead you to stories you would be likely to miss otherwise. For example, it is easy to imagine that the drug dealer and a woman with several boyfriends may each want to maintain multiple separate schedules in case the PDA is seen by the police or a boyfriend. The Pope probably has less need for secrecy but may want a larger font size.

So, while extreme characters may lead to new stories, it is hard to know whether those stories will be ones that should be included in the product. It is probably not worth much investment in time, but you might want to experiment with extreme characters. At a minimum, you can have a few minutes of fun thinking about how the Pope might use your software and it just may lead to an insight or two.

What If I Have On-Site Users?

The user role modeling techniques described in this chapter are still useful even if you have real, live users in your building. Working with real users will strongly improve your likelihood of delivering the desired software. However, even with real users there is no guarantee that you have the right users or the right mix of users.

To decrease the likelihood of failing to satisfy important users, you should do some simple role modeling on projects even when you have available internal users.

Summary

- Most project teams consider only a single type of user. This leads to software that ignores the needs of at least some user types.
- To avoid writing all stories from the perspective of a single user, identify the different user roles who will interact with the software.
- By defining relevant attributes for each user role, you can better see the differences between roles.
- Some user roles benefit from being described by personas. A persona is an imaginary representation of a user role. The persona is given a name, a face, and enough relevant details to make them seem real to the project members.
- For some applications, extreme characters may be helpful in looking for stories that would otherwise be missed.

Developer Responsibilities

- You are responsible for participating in the process of identifying user roles and personas.
- You are responsible for understanding each of the user roles or personas and how they differ.
- While developing the software, you are responsible for thinking about how different user roles may prefer the software to behave.
- You are responsible for making sure that identifying and describing user roles does not go beyond its role as a tool in the process.

Customer Responsibilities

- You are responsible for looking broadly across the space of possible users and identifying appropriate user roles.
- You are responsible for participating in the process of identifying user roles and personas.
- You are responsible for ensuring that the software does not focus inappropriately on a subset of users.
- When writing stories you will be responsible for ensuring that each story can be associated with at least one user role or persona.
- While developing the software, you are responsible for thinking about how different user roles may prefer the software to behave.
- You are responsible for making sure that identifying and describing user roles does not go beyond its role as a tool in the process.

Questions

- 3.1 Take a look at the eBay website. What user roles can you identify?
- 3.2 Consolidate the roles you came up with in the previous question and show how you would lay out the role cards. Explain your answer.
- 3.3 Write persona descriptions for the one most important user role.