the Rational edge
e-zine for the rational community

Features | Management | News | Rationally Speaking | Technical | Franklin's Kite | Reader Mail

subscribe
contact us
submit an article
rational.com
issue contents
archives
mission statement
editorial staff

What Makes a Good Use-Case Name?

by Dr. Use Case (aka Leslee Probasco)
Rational Software Canada

Dear Dr. Use Case,

We have a very short timeframe to review the artifacts for our upcoming release. We have contracted with a consultant to define our use cases (for an online auction site). I would like your help on use-case names ONLY. I am most concerned about the use case named "Auction." I have already provided the feedback that this might not be a good use-case name because it does not start with a verb, but we are going around and around with the fact that "Auction" is a verb. Proposal on the table is "Execute Auction," which does not seem very good to me. The whole e-commerce jargon is throwing me for a loop, so I am asking you (as a native English speaker) for feedback. Feedback by yesterday would be GREAT!!!! This thing is supposed to be available next Tuesday.

Thanks.

Tongue-Tied and Nameless

Dear Tongue-Tied,

I believe your concern about having clear and meaningful use-case names is very well founded; it's worth spending the time on up front to get them right. I'll use your questions about use-case names as a starting point to jot down some thoughts on this subject.

Why Should We Care About Use-Case Names, Anyway?

Why should we care so much about what name we give a use case? If we follow an iterative process, as the Rational Unified Process (RUP) recommends, then at the end of the project's Inception phase, the project team and customers must agree on scope definition and cost and schedule estimates. Ultimately they must make the decision to either proceed with the project or to cancel it (one of the objectives of the Inception phase).

Except for a few critical use cases that are outlined by the end of the Inception phase, the only information available about the identified actors and use cases are their names. Along with specified features and other system requirements, this must be sufficient for all stakeholders to have a clear enough understanding of the functionality of the system in order to make this critical "thumbs-up" or "thumbs-down" decision. This is made much easier, I believe, if clear and consistent guidelines are followed when naming the use cases. Often, in the early iterations of Elaboration, only a few use cases (those that are considered architecturally significant) are described in any detail beyond a brief outline.

A Rule-of-Thumb: Look at the Diagram

A basic rule-of-thumb I like to follow in naming use cases is that anyone (at least anyone familiar with the problem domain) should be able to look at a use-case diagram -- noting the actor and use case names, and their associations - and have a pretty good idea of the value or goal to be achieved by each use case. To accomplish this, it is very important to choose the names of all actors and use cases with this objective in mind.

The "Golden Rule of Use-Case Names": Indicate the Value or Goal

In the Guidelines for Use Cases, the RUP states that "each use case should have a name that indicates what value (or goal) is achieved by the actor's interaction with the system" (if all goes as expected! :>). I would call this the "Golden Rule of Use-Case Names."

Here are some good questions to help you adhere to this rule:

> *Why would the actor initiate this interaction with the system?*
> *What goal does the actor have in mind when undertaking these actions?*
> *What value is achieved and for which actor?*

Guideline: Use the Active Form

I agree with you, Tongue-Tied; my preference is to have a use-case name that begins with a verb. But why? The reason is that I believe the "Golden Rule of Use-Case Names" (to indicate the intended value or goal) is best achieved by stating the use-case name in the active form, beginning with a verb. This also helps maintain consistency across all use-case names in the project -- which may be written by different team members -- and helps avoid ambiguity.

Imagine a To-Do List

When naming use cases, I like to imagine the items I might write down on a to-do list. So, for example, if I were heading out to use an ATM system and I wanted to make a to-do list for what I hoped to achieve in my visit to the ATM today, I might write down such items as "Withdraw Cash," "Transfer Funds," "Service the ATM," "Deposit Funds," rather than "Cash Withdrawal," "Funds Transfer," "ATM Service," and the like.

Keep in mind, however, that this naming guideline that I've specified is not a hard and fast rule. Use whatever words promote the best communication with the customer. I have seen many use-case names that do not follow this guideline -- some are good, some are bad. Along the same lines, I've seen many use-case names that do follow this guideline -- some are good, some are bad. You get the picture!

Specify Project Conventions in the Use-Case Modeling Guidelines

To avoid confusion and ambiguity, it is best to set specific project guidelines (as you and your team are apparently doing now) so that your whole team will be consistent in naming all use cases for your project. Once this decision is agreed upon, your project's Use-Case Modeling Guidelines artifact is a good place to clearly re-state the "Golden Rule of Use-Case Names" (to indicate the intended value or goal), along with any preferences you would like to have applied; for example, the guideline of beginning each use-case name with a verb.

This is especially important if you are contracting with someone else to do the work and you won't be involved in the initial naming of the use cases. It will save you from many arguments over individual use-case names and will help ensure that the use cases are defined at the proper level of abstraction. Be careful, though, not to fall into the dreaded trap of "functional decomposition" of use cases. (See Kurt Bittner's article, "Why Use Cases are Not 'Functions'," in the December 2000 issue of *The Rational Edge*.)

Now, Let's Look at Your Use-Case Names

Let's assume that in the Use-Case Modeling Guidelines for this project you've noted: "All use-case names should indicate what value or goal is achieved by the actor(s)' interaction with the system and must be stated in the active form, beginning with a verb."

With this in mind, let's look at the current names you've chosen for the set of use cases in your project, NetMarket (an online auction Web site):

1. Abstract Ancillary Service
2. Auction
3. Catalog Management
4. Create Auction
5. Customer Service
6. Deploy
7. Edit Workflow

8. Edit Workflow Schema
9. Exchanges
10. Member Directory (profiles)
11. Order Fulfillment
12. Product Search Find
13. Reverse Auction
14. RFQRFP
15. Select ASP
16. Select EAI Provider
17. Set Dynamic Price
18. Vertical Communities
19. Workflow Process Automation

Based on your question to me, I will not delve into the questions of "Is this a good use case to have?" or "Couldn't this use case be combined with some other use case?" You'll need to work with your customer to make sure that each use case is actually intended to achieve a "real" value or goal. (I'm guessing that applying our guidelines for use-case naming will actually help us answer those questions, but we'll see. I haven't even looked at the contents of the use-case documents you sent me. At this point, I'm looking only at the names.)

Apply the "Begin with a Verb" Guideline

Keeping in mind my limited view of your project and domain, I can see that seven of the use-case names adhere to the guideline "must be stated in active form, beginning with a verb": 4, 7, 8, 13, 15, 16, 17. Still, I would encourage you to look closely at these to make sure they actually state an intended goal of the system. Are these items I might write on my "to-do" list if I were to head out to use this system?

Then you have to ask yourself: If only seven of the nineteen original use-case names conform to our desired guidelines for use-case naming, maybe it would have been better to clearly state this guideline up front before handing the project off to a contractor. (Anyway, you can add this item to your "Lessons Learned" list to avoid this problem in future projects.)

Fortunately, some of the remaining names should be easy to clean up and change to fit the naming rules we specified in our Use-Case Modeling Guidelines:

(3) Manage Catalog
(5) Get Service on a Problem
(11) Fulfill Order
(12) Search for a Product
(14) Request Quote (or Proposal)
(19) Automate Workflow Process

Evaluate the Results

So our set of use-case names now looks as follows (with "???" indicating those not yet resolved):

1. Abstract Ancillary Service -> ???
2. Auction -> ???
3. Catalog Management -> Manage Catalog
4. Create Auction
5. Customer Service -> Get Service on a Problem
6. Deploy -> ???
7. Edit Workflow
8. Edit Workflow Schema
9. Exchanges -> ???
10. Member Directory (profiles) -> ???
11. Order Fulfillment -> Fulfill Order
12. Product Search Find -> Search for a Product
13. Reverse Auction
14. RFQRFP -> Request Quote (or Proposal)
15. Select ASP
16. Select EAI Provider
17. Set Dynamic Price
18. Vertical Communities -> ???
19. Workflow Process Automation -> Automate Workflow Process

Go Back to the Customer to Resolve Questions and Verify the "Golden Rule"

Now we are down to six unresolved use-case names. I find some of these current names do not make it very clear exactly what you're trying to accomplish with each of the use cases. I would need to go back to the customer (or you, in this case) to really understand the goal you're trying to achieve with each one. This also points out why our naming guideline is a good idea. Let's look at the unresolved names one at a time:

(1) Abstract Ancillary Service -> ???
In this name, is "abstract" a verb or an adjective? Note, if we were following the guidelines, then I would know it was a verb, and that with this use case we are abstracting ancillary services. I have a feeling, though, that this is not really what you mean to say. Without the guideline in place, however, I would imagine that the system is providing some sort of ancillary service. In that case, a well-formed name might be "Request

Abstract Ancillary Service." Or maybe this is just a container for all the little things for which you don't want to write full use cases -- and that might be okay. Back to the stakeholder on this one...

Use Terminology That Fits the Domain

(2) Auction -> ???
This is the use-case name that you indicated in your email was causing some arguments: Is "*Auction*" used as a noun or a verb? If it is a verb, then a better name might be "Auction Item," or "Auction Product," or maybe even "Auction Off Item." However, without the object clearly specified, I may even wonder if this use case is for auctioning a group of items. By using our naming guidelines, we could avoid this confusion. We'd know both what the word *auction* means and exactly what is to be auctioned.

You mention a proposed name of "Execute Auction." Sounds to me like a software person proposed that name. I'm concerned that a person not familiar with software terms might think you were going to chop the head off the auction or put it in front of a firing squad! In this case, I'd check with someone familiar with the domain, like an auctioneer, to see what name would best fit. Also, who is getting the value here? The customer? If so, then perhaps the best phrase is "Participate in Auction" or "Bid on Item." Again, do you see how a well-formed name, along with the diagram showing the actor, would help us see this more clearly?

Don't Use Software Jargon

(6) Deploy -> ??? What is being deployed? My guess is that this might be a use case telling how to deploy the product. Again, it seems you are using terms from the software domain, rather than the customers' domain. If I were writing use cases on how to develop software, then "Deploy Product" might be a good choice. My guess is that it's not a good choice for your system, however. If it's a Web site that is being deployed, a better name might be "Launch Web site."

(9) Exchanges -> ??? Again, this is quite ambiguous. "Exchange Items"? "Exchange Money"? "Exchange Dance Partners"?

(10) Member Directory (profiles) -> ??? What are you doing to these? Perhaps "Update information in Member Directory (profiles)"?

(18) Vertical Communities -> ???
Sounds like you've got a proper domain term here. Now you just need to clarify what it is that is happening: the action. And what is the goal you're trying to achieve? In closing, I encourage you to get these questions answered and the issues resolved before you spend more time on writing detailed use cases.

Hope this helps. Let me know how it goes.

Usefully yours,

For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!