

# Expressing Requirements with User Stories

A big part of building a software system is determining what the customer wants. We call these requirements, and there are many techniques for eliciting or discovering requirements from a customer or user. If you want to learn more about the many ways in which to elicit and clarify requirements, you can check out the Software Project Management specialization provided by the University of Alberta and Coursera.

## User Stories

Once a requirement is elicited, it needs to be expressed in some form. One technique for expressing a requirement is called a **user story**. A user story is simply a requirement, often from the perspective of an end-user, which is stated in natural language. You may have seen Sam using one of these in the previous video. A user story looks like this:

As a \_\_\_\_\_, I want to \_\_\_\_\_ so that \_\_\_\_\_.

Put the *user role* into the first blank. This may be quite simple for software in which there is generally only one type of user, or a bit more complex in cases where the software may do different things for different people. Either way, this clarifies who wants to use this feature.

In the second blank, put that goal that the user role wants to achieve. This will lead to some feature that you want to implement.

After *so that*, put the reason why the user role wants this goal. Sometimes this clause is omitted if the benefits are clear and generally known.

After you fill in a user story, you can apply **object-oriented thinking** to it to discover objects and possibly further requirements!

## Example

Imagine that you introduce the user story tool to your client, and they give you the following sentence:

As an **online shopper**, I want to add an **item** to my **shopping cart**,  
so that I can purchase it.

Usually, the nouns correspond to objects in your software. So in this example, you have identified three objects: first, the user is associated with an object in the software (the online shopper). An item could be any product at the store, while a shopping cart is an object for storing items for purchase.

Let's have another look at the sentence:

```
As an online shopper, I want to add an item to my shopping cart,  
so that I can purchase it.
```

Verbs can help you identify the requirements that your objects might have. In this example, **add** and **purchase** might be responsibilities of the shopping cart. Verbs may also help you identify connections between objects.

The last point is a bit more subtle; a user story can also help you discover **connections** between objects. In this example, it is probably fairly obvious. One online shopper is typically associated with one shopping cart. The shopping cart should be capable of holding multiple items.

## Conclusion

**User stories** are just one of many techniques that can be used to express requirements for a software system. They are simple to use and can allow you to apply object-oriented thinking and discover objects and further requirements. Start using them today, even if the ideas are your own! By identifying objects before starting implementation, your software will become better structured and more clear.