**Writing Scenarios Guide**

Steps:

1. Learning goals
2. How to write a Test Scenario
3. Defining scenario success
4. How many scenarios to test?

***Learning goals***

Think about what you’ll want to be able to say you learned after your test sessions. These are the research goals you should make explicit before writing your user scenarios.

Some example questions your team might have:

* + - Has your team developed a new function and you want to test user understanding of it? [Text Wrapping Break]Can the user complete a task using this function?
    - Is your team debating the navigation items of a website application, and you’re not sure which is best?

***How to write a test scenario***

A scenario describes to the user what *actions you want them to take* using your prototype and should reflect *the learning goals* you want to accomplish.

A good scenario won’t focus on how a user would *use* a specific element of the User Interface (UI) but rather how a user would go about completing a *task* using the UI.

Your scenarios should provide some context of why a user would want to do a task so that they have enough motivation and direction to do it. It gives them a reason and a goal for completing the task, but it lets them show you how they would use the UI to accomplish that goal.

Generally, a scenario should be:

* + 1. **Realistic**: would this user type actually do this on their own using this [site/service]?
    2. **Actionable**: Your task should be written so that it sparks action. It's not enough to rely on what the user would tell you how they would complete the scenario. You need to watch them do it as they will probably do it differently.
    3. **Not biased**: Avoid using the labels and terms displayed on the UI uses when writing your tasks. Otherwise, you give clues as to what you may want the user to do. You’ll miss opportunities to see how a user would actually behave using the UI.

Things to watch out for when writing a scenario:

* + 1. **Don’t** write tasks that can give away the answer to what you want the user to do.
       1. For example: "Upload your grantee performance report using that button in the right corner."
    2. **Don’t** write tasks that provide too little instruction.
       1. For example: "Explore the home page."

Example of a good scenario:

*"You're interested in finding information on how to lower the cost of your monthly rent. Your friend told you that section 8 vouchers can help with this. Use this [website/web app] to find out if section 8 vouchers can help you lower the cost of your monthly rent."*

* + 1. This scenario is **realistic** because it involves using a website to look up information that the user cares about
    2. This scenario is **actionable** because it asks the user to use the website to find the pertinent information.
    3. This scenario **avoids bias** by not telling the user to specifically find the information.

***How many scenarios to test?***

User attention and ability to concentrate decreases after 30 minutes so your usability test sessions should run around that length. It’s helpful to run a pilot test of the tasks with a colleague to get a feel for how long it will take.

***Criteria for judging scenario success***

By defining ahead of time how you want to gauge success or failure for each scenario.

* + 1. Does this scenario help the user accomplish their goal?
       1. Example: the user was able to complete \_\_\_\_\_\_\_\_\_\_\_ task successfully.
    2. Does this scenario support discoverability?
       1. Example: the user recognized what the UI expected them to do.
    3. Does this scenario support navigation?
       1. Example: the user understands where they were in the system.
    4. Does this scenario support error avoidance and recovery
       1. Example: the user avoids filling out a form wrong because the non-necessary fields are grayed out.