# Checklist for UAT

After requirements and mockups are approved and before the developers are ready with a product ready for testing:

* In the project folder in Box, create a folder within Supporting Docs labeled Testing. Create a subfolder within that labeled Screenshots.
* Begin working from the Requirements document and any mockups or prototypes to develop the test script, using the \_ template as a guide. Keep in mind the following:
	+ Break down test script into the various roles that are described in the requirements. What should each set of users be able to accomplish?
	+ Users should have a set of tasks to accomplish. Each of these tasks can be broken down into individual steps. Steps should not be overly descriptive and are not to be considered the same as documentation – part of what we’re testing is the intuitive nature of the interface.
	+ Each task must map to at least one requirement for that role. Associated requirement(s) should be listed on the test script. (The requirements mapping should not be included in the final UAT script that testers see.)
* Consider the setup for testing. How many testers will be needed? What role(s) should those testers need? What will need to be prepared in terms of spaces and access? Document this in the first section of the UAT script and get feedback from the developers and the program manager, who might suggest changes based on technical limitations and constraints on resources.

When the developers indicate that the product is ready for testing:

* **Arrange with the developers to have an individual or small group from the UX team run through the test script to check for major issues** (this might be a process that happens iteratively as the product is being developed). Make sure that test spaces are set up and the various testing roles are covered. Document testing results along the way whether it is an iterative or all-at-once process. A recommended testing cycle might run like this:
	+ Stage 1: Lead developer & Lead Tester run test script
		- Deliverable: test script runs clean without issues, moves on to Stage 2
	+ Stage 2: Lead Tester and UX team members run test script, bug fixes go to developer, once fixed, bugs go to Stage 1 to be tested before moving on to Stage 3
		- Deliverable: test script runs clean without issues, moves on to Stage 3
	+ Stage 3: Lead Tester coordinates user testing group and session
		- bug fixes go to developer for resolution, once fixed, bugs go to Stage 1 to be tested and resolution confirmed
* When the product has passed initial testing round(s) with the UX team, it is ready for UAT testing.
	+ **Check for any upcoming blackout dates/times with the developers**.
	+ **Identify the individuals who will be participating in UAT testing and schedule their time as soon as possible (consult with PM if necessary).** Aim for conducting testing within two weeks from this point. The total time needed for testing is dependent on a number of factors. A general rule might be 2 hours for every page of actual testing steps for the staff role; 1 hour for each page of student/instructor role testing. This can vary quite a bit depending on the complexity of individual tasks/steps and is just a broad rule of thumb. Try to rate the complexity of tasks/steps and use information from the initial UX testing to judge what is needed. Then, aim for scheduling no more than one hour of testers’ time in a chunk, and break up tasks accordingly. Overestimate and schedule an extra session for each role – you may just have to cancel the extra sessions.
	+ **Schedule a room/rooms for testing.**
* **Schedule the testing on the developers’ testing calendar**.
* **Ask the developers to set up any test spaces per the agreed upon setup**. Add any sample content per the agreed upon setup.
* Do a final run-through of the test script with your test space, if necessary.
* Set up a Slack channel for recording issues during live testing, and add all testers.
* At least 24 hours before testing begins, remind attendees via email.
	+ Confirm their attendance.
	+ Make arrangements for remote access if necessary.
	+ Have testers confirm access to their space.
	+ Share the project proposal and requirements with the group.
* Have the UAT test recording grid set up with the. This step will save time during and after live testing as you record users’ browser/OS configurations and results.
* Print out hard copies of the test script, one for each tester plus the facilitator.
* Find out which developer will be standing by during live user testing, in case of major issues.

During testing:

* Start with a welcome message in the Slack channel and ask users to report their browser/OS combination (if it is not pre-assigned).
* As users work their way through the script, ask them to describe issues they encounter. The Slack channel is there for capturing screenshots, and any minor issues that testers don’t want to disrupt discussion for.
* Any major issues should be attempted to work out with the tester (it may be a usability issue or something that will need to be captured in documentation). Otherwise, in the case of a technical “show-stopper, contact the developer that you identified as being on standby.
* As much as possible, capture issues as they happen in the recording grid that you prepared.
* At the conclusion of testing, collect any notes that testers have made on the test script printouts.

After testing:

* Collect all physical notes as well as screenshots and notes that are in Slack, and add them to the recording grid.
* Notify the developers that the report is ready.