

Sleepio Onboarding Take Home Test Plan

Gather Requirements

First things first, I would gather the requirements from the product owner on the team to understand how this product should work and to help me prioritize what should be tested first. If there is acceptance criteria I will read and prioritize what's in there as it should give me a sense of what product wants out of the feature. I would also try to read any user stories to get a better sense of who is using this product. I would then also talk to the software engineer to get a better sense of how the feature or product is being built. I would then try to bridge the gap between product and engineering to see if there

Document workflows

The next thing I would do after gathering requirements is to document workflows and to create a spreadsheet of how many ways we can get through the system we are creating. In this take home assignment for instance I would calculate, then document, all the different ways the user can get through the questionnaire. I would then work with the product team to understand if any of the workflows were edge cases to better understand what should be prioritized.

In terms of documenting the workflows of the take home assignment and without having enough time to do a spreadsheet properly. From what I can gather, there are five main categories of score that the Onboarding Sleepio questionnaire can give, Very Poor, Poor, Fair, Good, and Excellent. The questionnaire also introduces additional questions based on the user's answers to questions about employment and sleep quality.

Input testing/exploratory testing

Beyond understanding and documenting the workflows. I would also test each individual inputs where a user can send keys. I would use this tool, the [Big List of Naughty Words](#) to help me determine if our inputs were sanitized (SQL and Javascript injection), accept special characters, and to see if there are certain string lengths that can break the inputs. I would also test for duplicates where applicable, like with email, where it's not appropriate to enter the same email twice as it would detect a returning user. This would also extend to the dropdowns to make sure that options outside of the actual options to make sure those were working as expected.

Further the Onboarding product also requires an answer to continue the questionnaire so I would see if inputting an answer then deleting it would trigger a false continue.

Automation

Using the 80-20 rule, where 80% of defects come from 20% of the software, I would try to make sure we had our main workflows covered (Very Poor, Poor, Fair, Good, and Excellent). I would also make sure that the additional questions that are triggered were also covered. I wouldn't try

to automate any defects from the input testing as that means we are probably missing a unit test. If that was the case, I would reach out to the engineer who developed that feature and work with them to get the missing unit test created.