---

Title: Test Driven UX

Date: 2019-4-5

Author: Tyler

mainImg: https://images.unsplash.com/photo-1489257712451-3a66755ca19c?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=700&q=80s

---

Test driven development has been around for 12 years and in the past ~3 it has really transformed from best practice to standard place. TDD transcends programming languages, coding best practices, and software development processes (waterfall, agile, XP, etc). The reason it continues to take over the world of software development is because it is scalable.

## What is test driven UX?

Test driven starts with a question/test. “What does the user need to be able to do that is currently either impossible, or needs made simpler?” Once the test is defined and agreed upon all future UX work is built around supporting that test (and all other out standing tests). Test driven UX has great recycling of effort with end to end testing or feature testing. To automate this process UX practitioners can learn tools like puppeteer or selenium and work with development to get these automated tests built into their deploy system.

## Why is this system beneficial?

At first TDUX may seem counter intuitive, there is a lot of overhead, and slows down executions.

* Writing clear requirements - Before jumping into design the first step is writing out requirements for expectations on users needs. This requires some conversations to be had upfront before pen ever hits paper.
* Reusable by QA - Since you are writing tests for things that users are able to do QA can take these items and implement them exactly as is.