Cash App

Client: Cash App Timeframe: 2 months Role: Director of Customer Success Ops Data Analysis Team: The Datapals (CS Ops Analytics) Methods: Feedback Surveys, Hallway Testing, Quantitative Tracking, and iteration Tools: Slack, Google Forms, Looker Time: 2020

Introduction: I walked into Cash App being told I should have an easy onboarding, and I should have known better because it was 2020.

Few if any of the reports we created as a team were being used regularly, and those that were often crashed, leading to failing user trust. Low levels of adoption were common among teams, leading to customized solutions with bad data and wasted efforts.

The goal of this study was to find out what was going wrong through UX research, specifically by creating open channel feedback, hallway testing, quantitative testing, and creating better feature release communication and schedules.

Methodology: Open Channel Feedback, Hallway Testing, Quantitative Testing, Metadata Analysis

Open Channel Feedback: The Datapals already had a Slack channel for users to reach out qith questions and report system outages. We widened this use to also ask for features feedback, and received a ton.

The biggest feature? Report uptime, particularly for our staff on the east coast.

Feature Tagging: Our feedback and communication channel was fantastic, but also very chaotic. We needed better numbers around the number and severity of outages, and to make it easier to navigate.

While this could be an entirely separate case study, our end result was to moderate the channel and force the use of threads by default for questions. As well, we normalized the use of emoji to signify a question had been handled, an outage had been reported, or a feature was requested.

Being able to count outages, and note what kind, helped us focus on the most important feature our users requested, report uptime. The visualization of feature requests also

provided transparency on why some feature requests were backlogged.

Metadata Analysis: Looker has a decent ability to track usage statistics, counting how many times a report had been viewed. By checking this list, we were able to see our popular reports, and develop a Retirement List of reports. Removing reports allowed us to focus on creating the polished experience our users told us they wanted, and relieved the pressure on our engineering staff.

Hallway Testing: Finding study participants can be interesting; finding ways to do hallway testing during a pandemic in a remote-only organization was a challenge.

In the end, we adopted the use of Office Hours to encourage people to drop in and ask us questions about our reports. During these office hours, we also asked for feedback on upcoming and in-progress designs.

We also started scheduling regular check-ins with our user base to make sure their needs were heard, and used these meetings for feature demos to ensure our ongoing progress was going in the right direction.

Findings: The findings from the open channel feedback and quantitative testing revealed several areas for improvement in the reporting system, including confusing navigation, removing unnecessary reports, and improving uptime.

We were able to priorize increasing uptime, and free up developer time by removing unnecessary report maintenance and nice-to-have feature requests, leading to a cleaner, faster reporting solution.

Reflection: UX research is for everyone making something. Internal reporting solutions are no less a product in need of research than an app or a device. Creating ongoing UX research as part of our report creation workflow helped us serve our customers better with each sprint.

In short, if data visualization is part of your job description, consider taking a UX/UI certification.