Respecting User’s Time
App developers want to provide their users with the best experience possible. Integral to the experience is how the app uses the user’s time.

Evidence of a poor experience is wasting the user’s time, which can happen in a variety of ways:
1. General slowness and lagginess.
2. A frozen screen forcing the user to tap on the screen multiple times for one action, like entering text.
3. Forcing the user to wait through visual cues, such as progress bars and throbbers.

These user-perceived errors indicate poor app Quality of Experience and can be measured to help developers improve the overall experience.

Quality of Experience (QoE)
Quantitative QoE is measured from the above events and other events that impact the user’s experience.

While Qualitative QoE is only estimated from user ratings and reviews, which is hard for developers to interpret at scale.

Debugging QoE
Replication of poor experiences is difficult because of device and version fragmentation in the Android world, and the developer may not know when the instance occurred.

Testing in their own environment provides limited data and insights into their apps’ QoE.

By modifying Android, developers can see how apps are behaving through their users’ eyes.

Progresso to the Rescue
Progresso is a system coupling Android platform instrumentation to log events that cause user-perceived latency—such as progress bars and the UI event queue—and an offline data analysis pipeline to quantify all apps’ QoE on a given Android smartphone.

Measuring Facebook
The figure below graphs the length of time, in seconds, a throbber was on the screen signaling the user to wait for a News Feed refresh.

In this instance, a poor QoE can be considered waiting for more than 1.0 second. This could be due to:
1. Network latency
2. Underpowered device
3. Interaction with other apps
4. Buggy or incorrect code
5. Other?

Other Approaches
Requires correct modification of apps.
May only run on developers’ devices.
Limited access to lower-level insights.
Harder to measure QoE-impacting events.

In-the-wild measurements on devices in different environments.
No developer work required.
Provides access to lower-level insights.
Requires a custom Android build.

Ongoing Work
Evaluation: Deploy to the 200+ participants in UB’s PhoneLab testbed and analyze apps at scale.