EXHIBIT A

Running Thoughtful Negative Tests
Desi Wang · Last edited January 2, 2020 · 11 minute read

Save

Running Thoughtful Negative Tests

Co-written with Xin Zhang

Negative test is a method that measures impact by intentionally degrading some user experiences. It provides a lot of insightful learnings in the realm of performance, but it is also a subject of controversy from time to time. We would like to share all the thoughts we put behind negative tests and things we feel important for running thoughtful negative tests.

WHY do we run negative tests?

An enduring topic that people who work on app and product performance study is the relationship between performance and user experience (measured by engagement, retention, perception, etc.). We would like to not just understand how user engagement and retention move directionally, but also quantify people's sensitivity to different performance changes. We have limited bandwidth to make improvements, and these learnings help us prioritize performance works and focus on the most crucial problems. Below are some common methods with their pros and cons.

Correlation analyses

The most convenient approach is to look at correlations. We can calculate correlation between any two metrics and quickly sanity check our hypotheses. However, correlations do not imply causation. Many confounding factors might take effects on both sides, and sometime the impact might even be the other way around. These analyses do not, on their own, tell us what would happen under a hypothetical change.

Example: Anna looked at the correlation between the latency of opening up permalink on Android and the likelihood of a user commenting on permalink. And she found that the probability of commenting actually increases when latency increases below 2s, and reaches peak at 2s. This is counterintuitive since we would expect latency and commenting rate have negative correlation. But there are many confounding factors: e.g. permalink with more total comments and less cached comments takes longer to load, but also has higher commenting rate (more confounding factors were discussed in her note).

Pre-tests and back-tests

Another approach that most people would agree on is to understand causal impacts through pre-tests and back-tests. There is no doubt that we should look at the engagement movements when we have performance wins in experiments and team holdouts, but we often find these not enough.

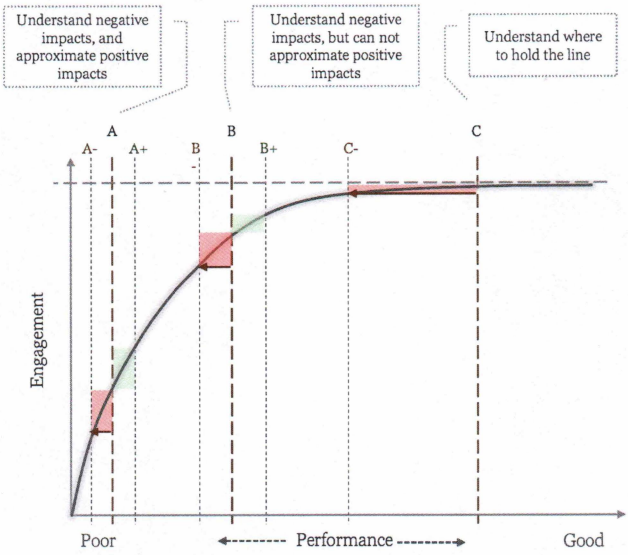
- Pre-tests and back-tests are not always available.
    - Some metrics are newly introduced without much learnings and some metrics might already be very optimal without many low-hanging fruits.
    - For some features and infra changes, back-test is technically impossible or hard to implement.
- The magnitude of performance wins in individual projects might not be big enough to trigger engagement movements. We need to be careful about drawing conclusions from results that are not statistically significant.
- It is very difficult to control the level of wins if we are looking for impacts in a certain range and/or systematic learnings on the relationships.
- We are not able isolate the impacts when multiple performance metrics benefit from the same change.

Negative tests

Negative test is an observational method with a lot of human controls. A well-designed negative test can provide systematic learnings on the relationship between a specific performance metric and user engagement. Combining with surveys, we can also understand how people's perception changes.

Performance and engagement often have non-linear relationships. We will likely see logarithmic curves where the marginal effects on engagement become smaller when performance improves. Depending on where along the curve we are, negative tests can provide us different learnings:

- Performance is poor and there is a lot of room for improvement (**A**): negative tests help us understand the negative impacts when moving from A to A-, and approximate the positive impacts when moving from A to A+.
- Perf improvements have smaller marginal effects than regressions (**B**): negative tests help us understand the negative impacts when moving from B to B-, but the positive impacts might be much smaller and hard to estimate.
- Performance is good and we are close to the horizontal asymptote (**C**): negative tests help us understand where to hold the line when we have limited bandwidth.



Depending on where along the performance-engagement curve we are, negative tests can provide us different learnings

Case study

In H2'19 we have both half-long holdout and negative test for *Feed image load performance*, and let us use this case to compare the pros and cons of the two methods.

Half-long Holdout	Negative Test
Pros	Pros
• Do not need to worsen user experience of image loading • Help understand how image load perf wins impact engagement like time spent and sessions • Learnings will be based on real image perf projects which can be used as a reference for future projections	• We can control image load delay latency and frequency and understand which one has more impacts • Help understand people's sensitivity to image load perf regressions • Help understand how perceived image load speed and image quality change by launching a survey along with the test • We can gather learnings quickly (in weeks)
Cons	Cons
• Need to hold for a long time and prevent some users from getting better image load perf • Learnings will be based on the actual size of image load perf wins during the half, which might be lower or higher than the level of impacts that we want to understand • Can not be used to understand how image load perf regressions impact engagement • Hard to separate impact of specific wins since overall effects might come from many experiments or multiple perf metrics • Some infra work might not be able to be gated and included in the holdout	• Can not be used to understand how image load perf improvements impact engagement • People have degraded image loading experience during test period, which can lead to long-term implications on engagement

Pros and cons of half-long holdout and negative test for Feed image load perf

However, negative test has a higher risk - it means real people will suffer from poor performance that they should not experience. Therefore, it is critical to think through how can we minimize the risk and run thoughtful negative tests.

HOW should we run negative tests?

Before a negative test

- **Study past learnings and ensure there is a need to run new negative tests.**
  - Negative tests are costly, we should avoid duplicate efforts if we can get enough guidance from past learnings. We will provide an overview of past learnings at the end of this note, please do check it out if you are planning for a negative test!
- **Choose the right target population to maximize the learnings and minimize the negative impacts.**
  - Choose the minimum test group size with needed statistical power.
  - *Filter by performance*: *Example*: Olga and the team ran a FB4A startup negative test to determine the value of a 3s vs 2.5s vs 2s startup time. The target population was selected from users with a p50 startup time of 2s, which allowed for measurement of the desired changes (2s vs. 2+500ms vs 2+1000ms) while avoiding unnecessary overexposure.
  - *Filter by logging*: Most of our performance logging is sampled from small portions of users. In order to understand impacts for users with different performance levels and join with survey data, we set the target population to include as many logged users as possible by choosing sampled users.
- **Choose reasonable regression sizes to be big enough to understand impacts and small enough to draw the red lines.**
  - Choosing big regression sizes might sound risky. However, if we cannot gather conclusive learnings from the test, we are putting our users through negative experiences for nothing.
  - On the other hand, we also want to choose some sizes that are small enough to understand where should we draw the red line. Testing extremely big regressions only provides less guidance to performance work because those regressions are unlikely to happen.
- **Design predictable and necessary negative experiences only.**
  - The actual experiences should match the regression sizes we chose and be predictable. *Example*: For Feed image load negative test, instead of injecting delays after images are rendered, we injected delays after the viewports are on screen. Because images will be rendered beforehand most of the time, and the actual delays seen on screen will not be predictable if we inject them right after rendering.
  - We should only design the test period for necessary amount of time, and minimize the risk of having long-term negative impacts on users such as permanent churns.
- **Communicate early and clearly with all stakeholders.**
  - Key things we should communicate before a negative test: our motivation, experiment set-up, timing, metrics to track and topline metric hit. If topline metric hit is unclear, we could estimate with employee dogfooding data.
  - We communicated our negative tests by posting experiment FAQs: iOS Comments Load Negative Experiment FAQ and iOS Feed Image Load Negative Experiment FAQ.

During a negative test

- **Closely monitor metrics and be prepared to turn off part or all of the test.**
  - Some test groups might have big regressions to ensure we have conclusive learnings. If one or more groups are showing worrying negative impacts on critical engagement metrics, we should make good judgements about whether we have enough data (from these or other test groups) to turn them off before planned date.
- **Communicate early results with the team.**
  - Communicating timely results can help the team set early expectations and lower anxiety. We also want to communicate the analyses planned for test data and survey results.

After a negative test

- **Closely monitor metric recovery.**
  - To understand long-term impacts, we need to closely monitor metric recovery or regressions after turning off the test. There are many good insights around obstinate effects and delayed effects. *Examples*: FB4A startup negative test shows sessions did not reach pre-experiment levels even after 2 months (post). FB4A scroll perf negative test shows MAP and WAP dropped a month after the experiment has been turned off (post).
- **Gather comprehensive learnings for engagement and user perception impacts.**
  - Critical *points*: People's sensitivity to performance regressions is typically inconstant when the level of regression changes. Understanding the critical points when negative experiences are noticeable, acceptable or unbearable is an essential need for negative tests.

- *Subpopulation*: People with different devices, network conditions and app performance might have different expectations and sensitivities to regressions. Finding out who are most affected helps teams to define at-risk cohorts and prioritize projects.
- *Downstream engagement impact*: Some downstream engagement impacts might show up unexpectedly and provide valuable insights. For example, Feed image loading experience was expected to mainly affect consumptions, but it also affected productions in our test due to less reshare.
- *Perceived performance*: Partnering with surveys, we can understand how people actually feel about the changes. And more importantly, we can validate whether our performance metrics are able to capture user perception and describe user experience accurately.

- **Make recommendations on team strategies.**
  - *What is the north star of the goal?* This is a fundamental question for all performance metrics. Negative tests help us understand whether we are good enough now, and the ideal stages of performance metrics (e.g. move long comments loading events into 2s, reduce 1s+ image loading events to below 2%, etc.).
  - *How should we construct investment portfolio among different performance metrics?* With limited resources, we often face decisions on which metric should be improved, which metrics should we hold the line, and which metrics can we budget for a regression, etc.. Negative tests provide us a way to systematically quantify impacts and make insightful decisions.

WHAT we have learnt from negative tests?

Facebook App

- **Image Perf iOS** (H2 2019): data and research
  - Slowing image loading experience has widespread user experience impact on consumption (VPV, Blue Time Spent, Video Time Spent), visitation (Sessions) and production (Reshare Broadcast Post).
  - Experiencing less frequent but slower outlier events is worse than experiencing more frequent but faster outlier events.
- **Comments TTRC iOS** (H2 2019): data
  - We recommend focusing on bad comments loading experiences and improve user-based metric Average % Bad (≥ 2s).
  - Comments load latency regression negatively impacts comment, like and overall Feed MSI.
- **Comments TTRC Android** (H1 2020): data
  - Comparing to the iOS test results, comments loading delays on FB4A have larger engagement impacts.
  - FB4A users are very sensitive to performance regressions on the Notifications surface.
- **App Startup Android** (H1 2019): data and research
  - We observed decreases in sessions and time spent supported the hypothesis of diminishing returns when moving from 3s → 2.5s vs 2.5s → 2s startup on the active DAP.
  - Perceived performance did not change significantly.
- **Touch Responsiveness** (H1 2019): data and research
  - Sessions, time spent, DAP, and MSI dropped significantly as soon as the regression was introduced, indicating that touch responsiveness highly impacts engagement metrics.
  - Perceived performance dropped across three most aggressive test variants.
  - Stall frequency appeared to significantly impact user engagement metrics. Certain high-frequency variants (500ms, 25%) had worse performance than high-magnitude variants (1000ms, 10%).
- **Stories Creation & Consumption** (H2 2018): data
  - Increased creation latency leads to regression in story production, and even consumption. Every 1s regression on creation latency can lead to ~0.2% loss in Story media posts and up to 0.1% loss in story producer DAP
  - Increased consumption latency leads to regression in story consumption. Every 100ms regression on server-side consumption latency can lead to 0.13% loss in Story consumer DAP and roughly 0.3% loss in story media views.
- **Responsiveness** (H1 2018): data

- Unresponsive delays significantly negatively affect engagement. Responsive ones are neutral. As a company, we should heavily favor moving work off the UI thread even if it results in a worse TTI.

- **Newsfeed Scroll Perf** (H2 2017): data
  - If we were to drop 100% more frames (= the net total of all regressions minus all improvements in ultrasound today), we will see
    - A 0.7% Time Spent and 1.4% VPVs drop on Newsfeed alone
    - A ~0.15% drop in MAP and WAP a month after we had turned off the experiment
    - A 2% drop in video TS in Newsfeed (inline) and 1.15% drop in overall video impressions.
- **App-wide scroll perf** (H1 2020): data
  - We recommend 25 LFD/m and 5 LFD/m as new scroll perf thresholds for "Bad" and "Great" in addition to the current "Good" threshold (15 LFD/m):
    - "Great": 5 LFD/m
    - "Good": 15 LFD/m
    - "Bad": 25 LFD/m

(coming soon)

- **iOS Feed & Stories Startup** (H1 2020)

Instagram

- App Cold Startup (H2 2016): data
  - Aggressive cold start reduction goals shouldn't be rationalized by major engagement wins. The 900 millisecond injection constituted a ~30% increase in cold start time and the biggest major engagement metric regression was comments, which fell ~1%.

FbLite

- App Code Startup (H2 2018): data
  - Startup-time regression harms core metrics, including visitation, time-spent and sharing
  - Dual-users visitation is compensated for in FB4A
  - Harm on core is more significant for high-end devices
  - Startup regression significantly increases <=1 second DAP

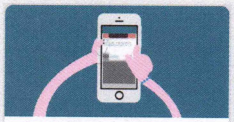
Thanks

Thanks to Di Lu, Olga Gritsevskaya, Anna Khasanova, Michael Shaw, Michael Midling, Selig Davis, Mike Plumpe, Itai Rosenberger, Oliver Rickard, Justin Coughlin, Diego Carranza, Jack Li, Luis Gomez, Karthik Veeramani for the amazing work in performance world and being supportive for running negative tests (sorry if we miss anyone). Special thanks to Jason Wei and Zuzka Bodik for providing feedback to this note.

Appendix

- Performance and Engagement Studies at Facebook Wiki by Vikram Rao, Michael Midling and Jerrod
- Negative Tests Working Doc by Desi and Xin

Recommended Reading


iOS Feed Image Load Negative Experiment Final Results
Desi Wang

iOS Story Viewer Negative Experiment Results (Initial Load an...
Yue Wang


FB4A Startup Negative Experiment: Final Results
Olga Gritsevskaya