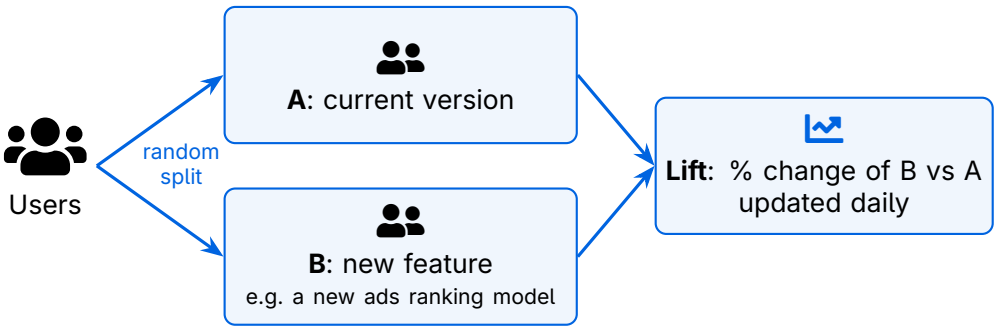
Let Them Peek:
Group Sequential Designs for A/B Testing

Lorenzo Perini, Mia Garrard, Rina Friedberg, Rohit Kumar,
Houssam Nassif, Abbas Zaidi, Pablo Alcain

Meta Platforms, Inc.

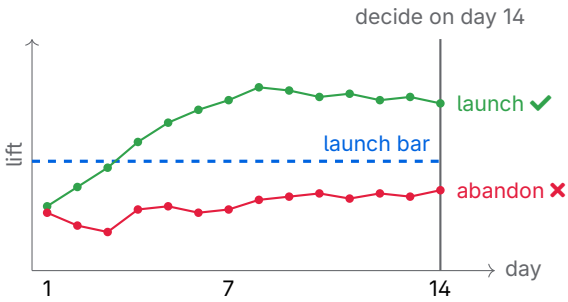
CODE@MIT 2026

To Decide Whether to Launch a New Feature, Teams Run A/B Tests



❓ Is the new feature better? Do we **launch** it or **abandon** it?

Today, Teams Launch Only If the Day-14 Lift Clears the Launch Bar



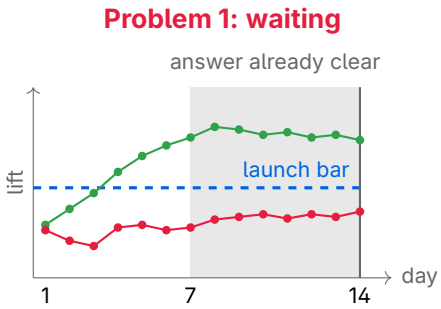
📅 Decide only on the **last day**

🚀 Launch if the lift is above the **launch bar**

⚖️ **False positive rate (FPR)**: at most 10% of no-effect features launch

🎯 **Power**: 80% of target-size effects launch

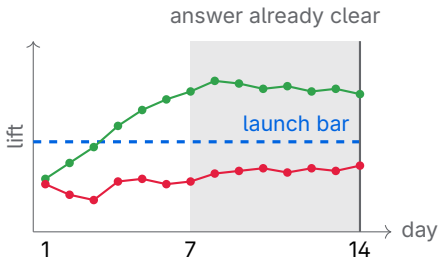
Waiting Wastes Scarce Slots, but Peeking Causes False Launches



⏳ Obvious winners and losers must still be tested until day 14, wasting resources

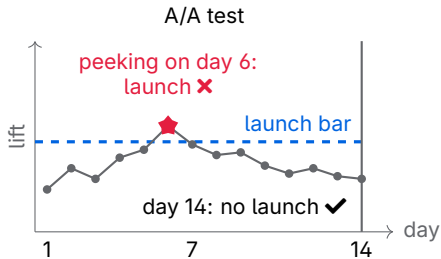
Waiting Wastes Scarce Slots, but Peeking Causes False Launches

Problem 1: waiting



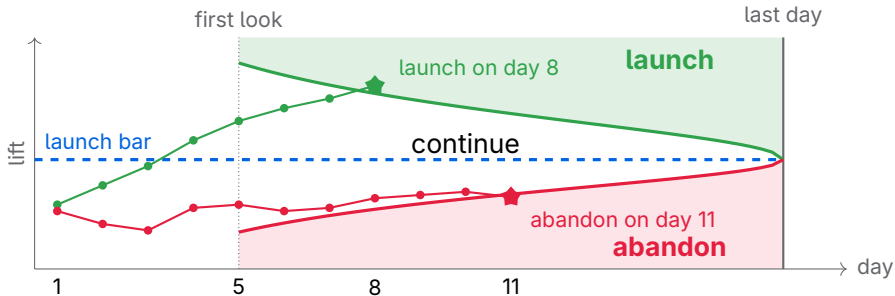
⏳ Obvious winners and losers must still be tested until day 14, wasting resources

Problem 2: peeking



👁 Teams save time stopping on the first day the lift crosses the bar ⇒ more **false launches**

A Group Sequential Design Stops as Soon as the Answer Is Clear



- The same two experiments as before are decided on days 8 and 11 instead of 14

To Be Used in Practice, the Design Must Meet Four Requirements

① 

Teams want to maintain the same terminal rule

② 

Readings from the same users are autocorrelated

③ 

Measure FPR and power without ground truth

④ 

Use as few days as possible

❓ How do we build a design that meets all four?

Methodology

Every day, our GSD policy monitors the lift and stops as soon as the answer is clear, while:

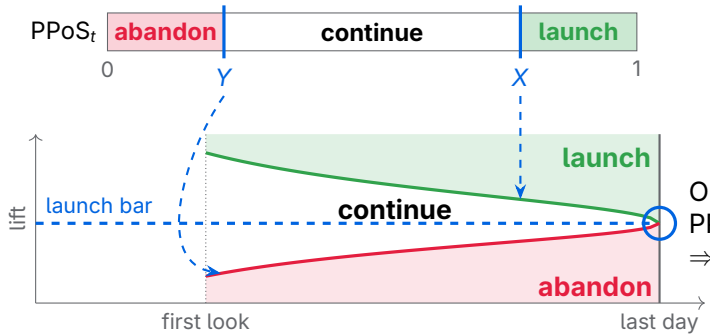
① matching today's launch rule on the last day
② accounting for the correlation between days
③ keeping FPR and power on target
④ minimizing the expected duration

① Thresholding PPoS Keeps the Same Terminal Rule

$$\text{PPoS}_t = \text{probability that } \hat{\theta}_T > \theta^\star \text{ given the data up to day } t$$

- PPoS_t: predictive probability of success on day t
- $\hat{\theta}_T$: lift on the last day T
- $\theta^\star$: launch bar

① Thresholding PPoS Keeps the Same Terminal Rule



On the last day,
PPoS$_t$ is 0 or 1
⇒ **same terminal rule**

② We Count Effective Days, as the Same Users Return Every Day

If each day brought new users



each day adds a full day of information

In reality the same users return



each day adds less: days are correlated

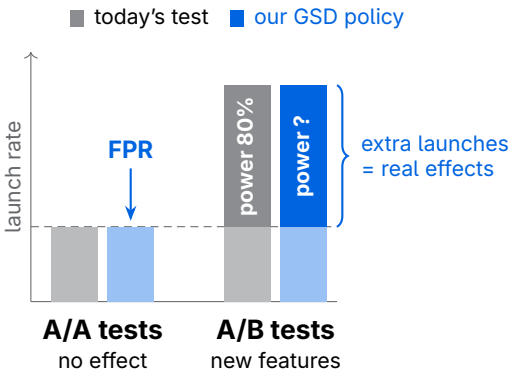
t: calendar days

$$n_{\text{eff}}(t) = t^2 \,/\, \sum_{i,j \le t} R_{ij}$$

effective number of days after t calendar days

correlation between days i and j, summed over all pairs, measured on A/A tests

③ Two Estimators Check FPR and Power Without Ground Truth

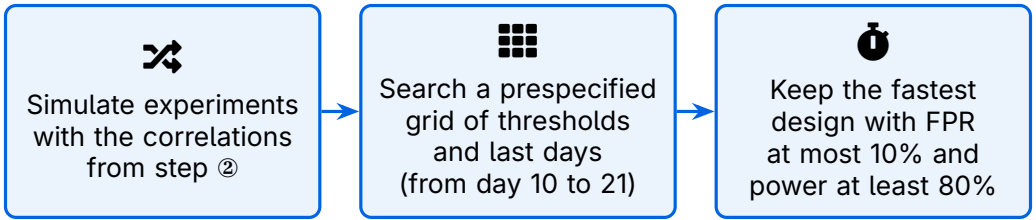


Two estimators

⚖ **FPR**: our launch rate on A/A tests, where every launch is false

◎ **Power**: anchored to nominal 80%, scaled by the ratio of extra launches

④ We Tune the Thresholds and Last Day to Minimize Expected Days



- Chosen last day: **17**, three days after today's 14
- The chosen design then runs on online experiments **without re-tuning**

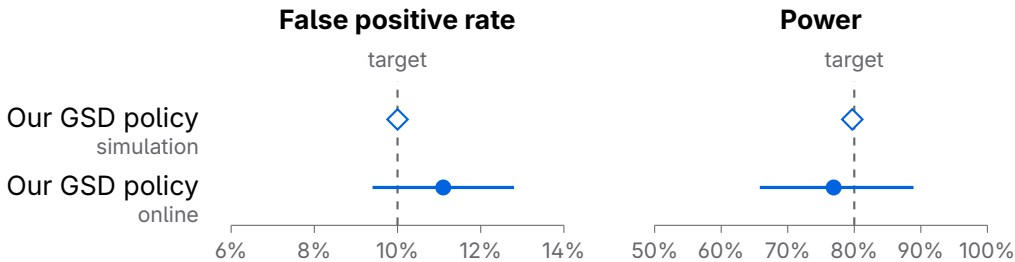
Experimental Results

Q1. Does our GSD policy keep FPR and power on target?

Q2. How many days does it save?

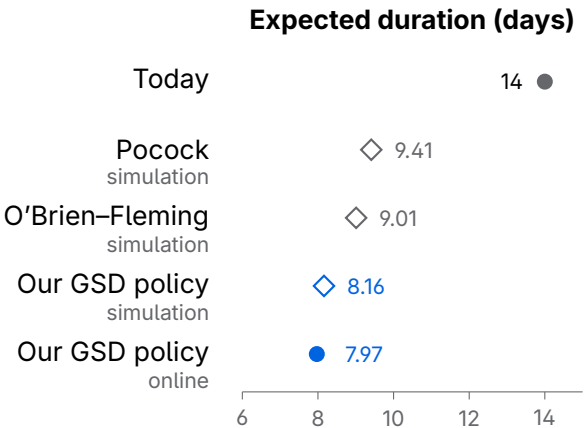
Q1. Our GSD Policy Keeps FPR and Power on Target

About 2,000 online A/A and A/B experiments on a supply metric



Online, both 95% intervals (bars) contain their targets, as in simulation

Q2. Our GSD Policy Cuts the Expected Duration by 43%

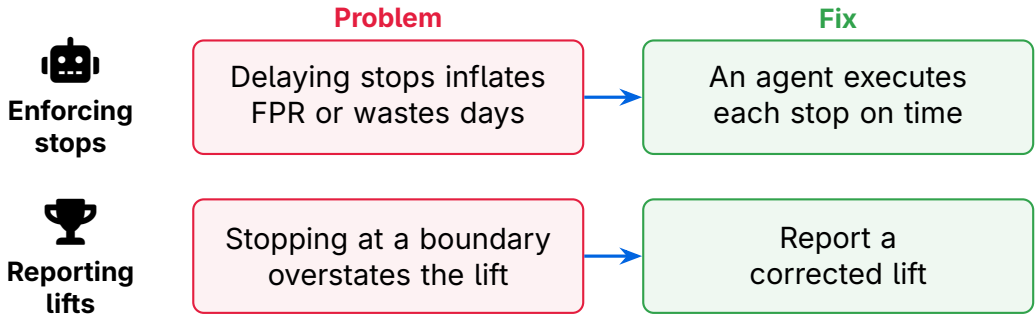


−43%

days on online experiments
(7.97 instead of 14)

Pocock and O'Brien–Fleming, two classical GSDs, are slower (O'Brien–Fleming even with the same last day, 17) and do not keep today's launch rule ①

In Practice, Stops Must Be Enforced and Lifts Corrected



Our GSD Policy Cuts Duration by 43% with FPR and Power on Target

① Same terminal rule on the last day

② Autocorrelation handled by effective days

③ FPR and power on target, without ground truth

④ 43% fewer days

lorenzoperini@meta.com

Meta