screenshotbot

Scaling up your Screenshot Tests

(without the Friction!)

Arnold Noronha
arnold@screenshotbot.io

Oct 9th, 2026

Founder, Screenshotbot

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

Quick refresher

- What is screenshot testing?
- Well known libraries
  - `Paparazzi for Android`
  - `swift-snapshot-testing for iOS`
  - `Golden Tests for Flutter`
  - *... and more*
- Convert all of your previews into screenshot tests!

screenshot-test or *snapshot-test*?

- Android devs call it *screenshot* (usually)
- iOS devs call it *snapshot*
- Blame me:
  `github.com/facebook/screenshot-tests-for-android` (2015)
- Flutter calls it *"golden tests"*, Web calls it *"visual regression test"*

screenshot-test or *snapshot-test*?

- Android devs call it *screenshot* (usually)
- iOS devs call it *snapshot*
- Blame me:
  `github.com/facebook/screenshot-tests-for-android` (2015)
- Flutter calls it *"golden tests"*, Web calls it *"visual regression test"*

screenshot-test or *snapshot-test*?

- Android devs call it *screenshot* (usually)
- iOS devs call it *snapshot*
- Blame me:
  `github.com/facebook/screenshot-tests-for-android` (2015)
- Flutter calls it *"golden tests"*, Web calls it *"visual regression test"*

screenshot-test or *snapshot-test*?

- Android devs call it *screenshot* (usually)
- iOS devs call it *snapshot*
- Blame me:
  `github.com/facebook/screenshot-tests-for-android` (2015)
- Flutter calls it *"golden tests"*, Web calls it *"visual regression test"*

screenshot-test or *snapshot-test*?

- Android devs call it *screenshot* (usually)
- iOS devs call it *snapshot*
- Blame me:
  `github.com/facebook/screenshot-tests-for-android` (2015)
- Flutter calls it *"golden tests"*, Web calls it *"visual regression test"*

Screenshot Testing—and AI

- UI code is *low-risk* for AI
- Screenshot tests let you move faster with AI
- Screenshot tests are also super easy to write with AI (mock data etc.)
- Keeps *vibe-coding* in check

Screenshot Testing—and AI

- UI code is *low-risk* for AI
- Screenshot tests let you move faster with AI
- Screenshot tests are also super easy to write with AI (mock data etc.)
- Keeps *vibe-coding* in check

Screenshot Testing—and AI

- UI code is *low-risk* for AI
- Screenshot tests let you move faster with AI
- Screenshot tests are also super easy to write with AI (mock data etc.)
- Keeps *vibe-coding* in check

Screenshot Testing—and AI

- UI code is *low-risk* for AI
- Screenshot tests let you move faster with AI
- Screenshot tests are also super easy to write with AI (mock data etc.)
- Keeps *vibe-coding* in check

Screenshot Testing—and AI

- UI code is *low-risk* for AI
- Screenshot tests let you move faster with AI
- Screenshot tests are also super easy to write with AI (mock data etc.)
- Keeps *vibe-coding* in check

Why teams give up on screenshot tests

- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Why teams give up on screenshot tests

- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Why teams give up on screenshot tests

- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Why teams give up on screenshot tests

- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Why teams give up on screenshot tests

- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Why teams give up on screenshot tests

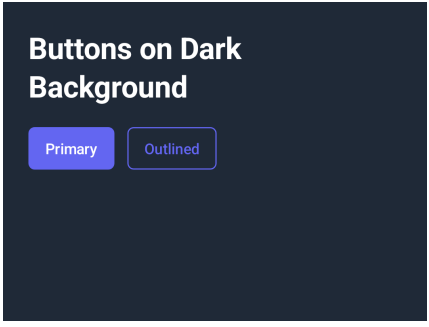
- **Education:** Everyone needs to know how to re-`record`
- **Storage:** where do you store the screenshots?
- **Friction:** manually re-`record` screenshots
- **"Flakiness":** laptops might not match CI machines
- **Justify** all this overhead!

Education

- Teach everyone `record`/`verify`
- E.g., if you add screenshot tests on a product screen, design system engineers need to know how to re-`record` *your* screenshots!
- ... we're not off to a good start, getting consensus is hard

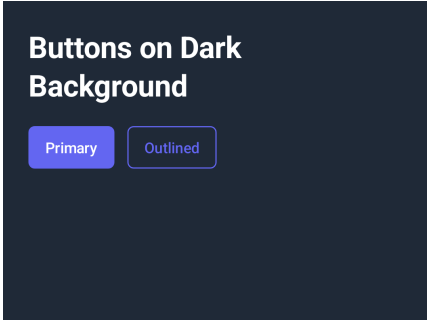
Design System teams vs Product teams

Design System...

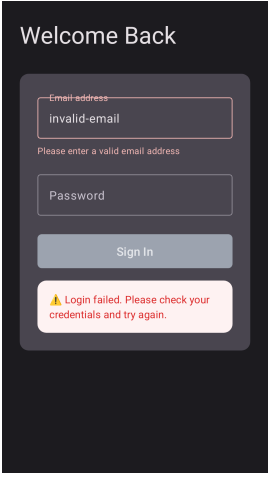


Design System teams vs Product teams

Design System...



...vs Product



The importance of Image Comparison

We'll do it live!

Friction

- A UI developer's job is to make UI changes
- They are re-recording screenshots multiple times a day, or waiting for Agents to regenerate it
- *"I'll wait for my screen to become stable before adding screenshot tests"*
- ... we can't **organically** scale up

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- … across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is 2 GB
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- ... across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is 2 GB
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- ... across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is 2 GB
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- ... across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is 2 GB
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- ... across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is $2\,\text{GB}$
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

Storage considerations

- A single screenshot is typically $\sim 50\,\text{kB}$
- 1000 screenshots are $\sim 50\,\text{MB}$
- ... across 100 UI changes total $\sim 5\,\text{GB}$
- GitHub's max free repository size is 2 GB
- Git LFS can help, but adds DevOps overhead

We don't want developers spending time figuring out which screenshot tests are *more* important!

“Flakiness”

- Screenshot tests are very stable if it runs on identical machines
- But Android developers use MacBooks, and CI runs Linux
  - `record` happens on dev MacBook
  - `verify` happens on Linux CI
- Even between Macs, there’s rendering differences between M1 and M4, and even different MacOS versions
- Flakiness causes people to lose confidence in the tests

“Flakiness”

- Screenshot tests are very stable if it runs on identical machines
- But Android developers use MacBooks, and CI runs Linux
  - `record` happens on dev MacBook
  - `verify` happens on Linux CI
- Even between Macs, there’s rendering differences between M1 and M4, and even different MacOS versions
- Flakiness causes people to lose confidence in the tests

“Flakiness”

- Screenshot tests are very stable if it runs on identical machines
- But Android developers use MacBooks, and CI runs Linux
  - `record` happens on dev MacBook
  - `verify` happens on Linux CI
- Even between Macs, there’s rendering differences between M1 and M4, and even different MacOS versions
- Flakiness causes people to lose confidence in the tests

“Flakiness”

- Screenshot tests are very stable if it runs on identical machines
- But Android developers use MacBooks, and CI runs Linux
  - `record` happens on dev MacBook
  - `verify` happens on Linux CI
- Even between Macs, there’s rendering differences between M1 and M4, and even different MacOS versions
- Flakiness causes people to lose confidence in the tests

Why not use thresholds?

- Decision paralysis: what's a good threshold? 99%? 99.99%?
- Might miss real changes in full-screen screenshots
  - e.g. rounded borders $\rightarrow$ square borders
- High support overhead
  - *"Why am I getting this notification, the screenshots look the same to me!"*
  - Look at the GitHub issues for any screenshot testing library

Justify

- Screenshot tests are the only kind of tests that everybody understands—designers, PMs, managers
  - ... It should be a way of convincing people to write more tests
- Why aren't we sharing these screenshots more?
- Can we export data to show the kind of bugs we're catching?
  - Screenshot tests changes frequently, which gives us a lot of data to work with

Justify

- Screenshot tests are the only kind of tests that everybody understands—designers, PMs, managers
  - ... It should be a way of convincing people to write more tests
- Why aren't we sharing these screenshots more?
- Can we export data to show the kind of bugs we're catching?
  - Screenshot tests changes frequently, which gives us a lot of data to work with

Justify

- Screenshot tests are the only kind of tests that everybody understands—designers, PMs, managers
  - ... It should be a way of convincing people to write more tests
- Why aren't we sharing these screenshots more?
- Can we export data to show the kind of bugs we're catching?
  - Screenshot tests changes frequently, which gives us a lot of data to work with

The solution

TL;DR

TL;DR

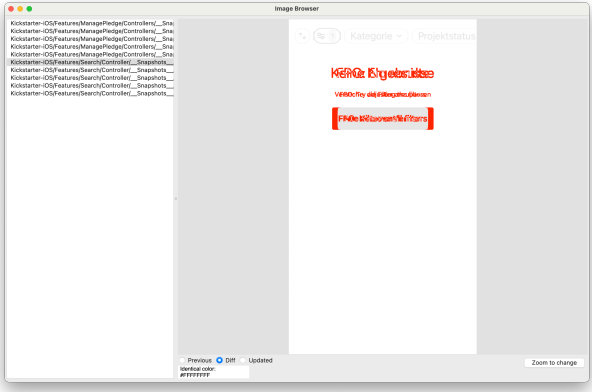
`record` on CI!

You don’t have to catch every regression

Prioritize Developer Productivity

Alternative Image Comparison tool

- Try Pixel-Diff, an open-source desktop image comparison tool (add it to a skill!)



Disable truly flaky tests

- If you accidentally push a flaky screenshot test, you need a way to disable the test. Reduce noise for the team!
- Hard to implement without a cloud solution, `verify` can't happen during the tests

Storage

- Git LFS is better than plain Git, ...
- S3 scales better, ...
- But dedicated cloud solutions for screenshot testing can better optimize build times
  - avoids large clones for each and every CI build

Turn screenshot testing into a collaboration tool

- Set up notifications when screenshots change. Designers, PMs *want* to see this!
- Increases visibility into testing efforts
- Minor: also track Accept/Reject rates.
  - Be aware that Reject rates tend to be low
  - ... but the rejected reports are valuable signal!

Where do you go from here

- If you don't already have screenshot tests, maybe add one today
  - Don't wait for consensus, issues can be solved later!
- Either build custom internal tooling...
- ...or try out something like Screenshotbot

Where do you go from here

- If you don't already have screenshot tests, maybe add one today
  - Don't wait for consensus, issues can be solved later!
- Either build custom internal tooling...
- ...or try out something like Screenshotbot

Where do you go from here

- If you don't already have screenshot tests, maybe add one today
  - Don't wait for consensus, issues can be solved later!
- Either build custom internal tooling...
- ...or try out something like Screenshotbot

Where do you go from here

- If you don't already have screenshot tests, maybe add one today
  - Don't wait for consensus, issues can be solved later!
- Either build custom internal tooling...
- ...or try out something like Screenshotbot

Quick word about Screenshotbot

screenshotbot

- Open-source and actively developed!
  `https://github.com/screenshotbot/screenshotbot-oss`
- Solves the storage issues we talked about
- PR notifications for devs, and Slack notifications for everyone else
- Figma integration so people can see what the designers intended
- History of every screenshot, with the ability to bisect regressions
- Collaboration tools for Designers, PMs, cross-platform devs, QA

Resources, Links, and Slides

- Links to these slides, and all referenced open-source libraries
- Feel free to modify the slides to present to your team!



screenshotbot.io/resources/nextapp26

Thank you

Questions?



screenshotbot.io/resources/nextapp26