centercode

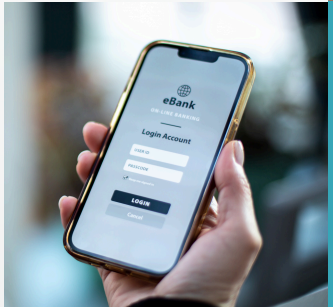
THE RELEASE PLAYBOOK

PREPARING YOUR APP FOR BETA









Chaos to Confidence

Getting a product out the door can be chaotic, but let's be fair—so is building one. The only difference is that when you're pushing that release to your market, you inherit some risk. These include risks like releasing a buggy product, missing functionality customers expect, and not meeting market needs. So, we invest in processes and programs to mitigate that risk. One such investment is beta testing.

Beta testing can feel a bit chaotic too. Testers are eager to fill your team's backlog. Developers are anxiously—and begrudgingly—anticipating a deluge of their mistakes. Meanwhile, stakeholders are impatiently sending Slack messages asking, "Is it ready yet?"

As someone responsible for the release, you're in the middle, keeping everyone happy (or at least trying to) while ensuring a functioning app gets into testers' hands as smoothly as possible.

Here's the thing: beta is supposed to be messy. It's when apps get pushed, pulled, poked, and prodded to iron out the kinks. But the process of rolling out a beta build to external testers? That doesn't have to be as messy.

With over 20 years of experience and thousands of beta tests under our belt, Centercode helps companies turn user feedback into actionable insights. In that time, we've helped hundreds of companies launch pre-release versions of their apps for beta testers.

This guide for release managers offers a practical overview of what to do to prepare for your app's beta release. We hope the tips in this resource help ensure every update lands smoothly, engages your testers, and keeps your team aligned. **Let's turn potential chaos into complete control!**

Table of Contents

PREPARING YOUR APP FOR EXTERNAL TESTERS 4

- Technical Assets ____ 5
- Informational & Visual Assets ____ 6
- Distribution Assets ____ 7
- Feedback Collection Tools ____ 8
- Communication & Marketing Assets ____ 9
- Risk Mitigation & Rollback Plans ____ 10

FINAL THOUGHTS 11

- Why Centercode ____ 11

APPENDIX 12

- Template: How to Install TestFlight and Join a Beta Test ____ 12
- Template: How to Join a Beta Test on Google Play Store ____ 13

Preparing Your App for External Testers

This guide focuses on the practical steps and resources needed to prepare your app for external beta testers. From builds and visuals to distribution and communication, **here's what you need to ensure testers can evaluate your app effectively:**



TECHNICAL ASSETS:

These components are required to ensure the app is stable, functional, and compatible for beta testing.



INFORMATIONAL & VISUAL ASSETS:

These materials provide essential details about the app and enhance testers' understanding of its purpose and functionality.



DISTRIBUTION ASSETS:

Tools, configurations, and resources used to distribute the app and manage the beta process.



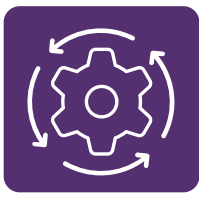
FEEDBACK COLLECTION TOOLS:

Systems and processes for gathering and organizing feedback from testers.



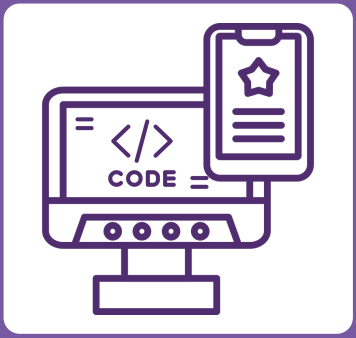
COMMUNICATION & MARKETING ASSETS:

Resources to inform, guide, and engage testers throughout beta testing.



RISK MITIGATION & ROLLBACK PLANS:

Strategies and plans to address potential issues during beta testing.



Technical Assets:

Setting the Foundation

Your app's technical assets are the backbone of the beta testing process. They ensure your app is stable, functional, and compatible across various devices and operating systems. Let's get these ducks in a row.

HERE'S WHAT YOU NEED TO PREPARE:

Signed app builds

- For Android, ensure you have a signed APK or App Bundle. For iOS, your IPA file must be signed with a valid provisioning profile and distribution certificate. No unsigned builds, no exceptions.

Configuration files

- Ensure any necessary configuration files (e.g., google-services.json for Android or GoogleService-Info.plist for iOS) are included in your build. These files are essential when using services like Firebase.

PRO TIP:

RUN A FINAL SMOKE TEST **BEFORE YOU DISTRIBUTE THE BUILD** TO CONFIRM THERE ARE NO GLARING ISSUES.

Informational and Visual Assets:

Communicating Clarity

Informational and visual assets are your app's first impression. These assets help testers understand your app's purpose, functionality, and what you're hoping to learn from the beta.

HERE'S WHAT YOU NEED TO PREPARE:

App description

- Write a brief, clear description explaining what your app does, its key features, and what testers should focus on during the beta.

EXAMPLE

"[App Name] is a fitness tracker designed to make achieving your health goals effortless. Key features include personalized goal setting, daily progress tracking, and workout reminders. During this beta, we're looking for feedback on the accuracy of tracking and usability of the reminders."

Screenshots

- Include high-quality screenshots that show off your app's main features. Make sure they meet platform-specific requirements (e.g., iOS screenshots for different device sizes).

Privacy policy & terms of use

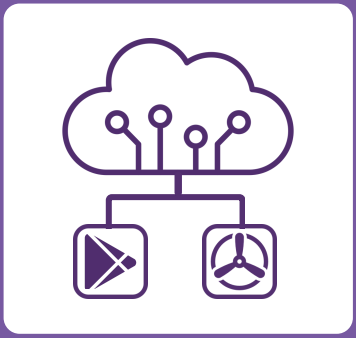
- If your app collects user data, link to a privacy policy. Platforms like TestFlight and Google Play Console require this, so don't skip it.

Feature graphics or videos (optional)

- A visually engaging feature graphic or a short demo video can make your app stand out and help testers quickly understand its value.

RESOURCE:

We've created this custom GPT to craft clear, engaging beta app descriptions for TestFlight and Google Play Beta, helping developers share key details while following platform guidelines.



Distribution Assets:

Getting the App to Testers

Distributing your app to beta testers should be a frictionless experience. These assets ensure testers can access, download, and install your app without hassle.

HERE'S WHAT YOU NEED TO PREPARE:

Platform configurations

- **For iOS:** Use TestFlight to upload your IPA, configure your tester group, and set expiration dates.
- **For Android:** Use Google Play Console to set up Internal, Closed, or Open testing tracks and upload your signed APK or App Bundle.

Tester onboarding instructions

- Provide clear instructions for testers on how to download and install your app. Spell it out—don't assume they know where to click.

TEMPLATE RESOURCES

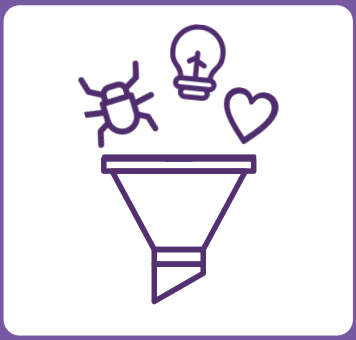
Discover onboarding instruction templates for testers on iOS & Android in the appendix.

Access management

- Decide how testers will access your app. Use email-based invitations for TestFlight or closed group enrollments in Google Play Console to maintain control over who's testing.

PRO TIP:

TEST THE ENTIRE DISTRIBUTION PROCESS YOURSELF TO **ENSURE THERE ARE NO HICCUPS FOR YOUR TESTERS**.



Feedback & Data Collection Tools:

Gathering the Good Stuff

Your feedback collection tools gather tester feedback and turn those experiences into actionable insights.

A good system ensures you can capture, organize, and prioritize their feedback.

HERE'S WHAT YOU NEED TO PREPARE:

In-app reporting

- Add a simple way for testers to report bugs or issues directly from the app. Include a screenshot or log-capture option to make their reports more detailed.

Crash reporting

- Integrate tools like Firebase Crashlytics or Sentry into your app to capture crashes and errors in real time. These tools give you the logs and stack traces that developers love and testers can't provide.

Issue tracking

- Use tools like Centercode to collect issues, categorize them (e.g., critical, minor), and prioritize fixes. Assign clear owners for each issue to ensure a closed feedback loop.

Surveys

- Prepare surveys with targeted questions to gather structured feedback. Use tools like Google Forms, Typeform, or Centercode to make this easy for testers and actionable for you.

Performance monitoring

- Set up tools to track app performance metrics like load times, battery usage, and memory consumption. This helps you spot inefficiencies before testers complain about them.

PRO TIP:

KEEP FEEDBACK CHANNELS SIMPLE AND FOCUSED. OVERLOADING TESTERS WITH TOO MANY OPTIONS CAN DISCOURAGE PARTICIPATION.



Communication & Engagement Assets:

Keeping Testers in the Loop

Communication is the glue that holds your beta program together. It ensures testers feel supported, engaged, and aligned with the goals of your beta.

HERE'S WHAT YOU NEED TO PREPARE:

Welcome email

- Send a warm, informative welcome email to all testers. This sets expectations and provides clear instructions to get started.

Regular updates

- Keep testers engaged by sharing updates about new builds, fixed issues, and what to test next. Regular communication reminds testers that their feedback matters.

TEMPLATE RESOURCES

For a **welcome email template** and more beta tester communication tips, check out the Tester Communication Playbook.

Get the playbook



FAQs

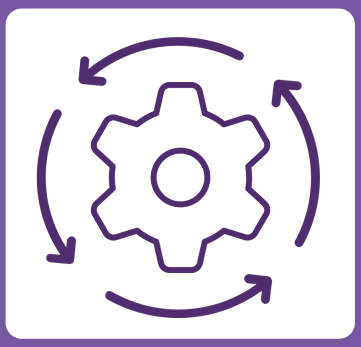
- Create a document or webpage answering common questions about the beta. Address installation issues, reporting bugs, and key deadlines.

Community channels

- Set up a space where testers can share feedback, ask questions, and collaborate. Tools like Slack, Discord, or forums can work well. Keep a moderator or team member active to manage conversations.

PRO TIP:

RUN A FINAL SMOKE TEST **BEFORE YOU DISTRIBUTE THE BUILD** TO CONFIRM THERE ARE NO GLARING ISSUES.



Risk Mitigation & Rollback Plans:

Preparing for the Unexpected

Things can and will go wrong during beta testing. Having solid risk mitigation and rollback plans ensures you're ready to address issues without derailing the program.

HERE'S WHAT YOU NEED TO PREPARE:

Rollback plan

- Always have a previous stable build ready to revert to if the current beta build causes major issues. Ensure distribution tools like TestFlight or Google Play Console can handle quick build swaps.

Hotfix workflow

- Define a clear process for efficiently identifying, developing, and deploying hotfixes. Be prepared to release these quick fixes for critical issues reported during beta.

Contingency plan

- Outline what the team will do in case of severe problems like app crashes, server outages, or data corruption. Be sure to include steps for internal escalation and external communication with testers.

Risk assessment checklist

- Proactively identify and address potential risks before the beta begins:
  - Have you tested the app on all target devices?
  - Are all critical features functioning as intended?
  - Do you have a backup of server configurations and data?

PRO TIP:

KEEP COMMUNICATION TRANSPARENT. IF A MAJOR ISSUE ARISES, **INFORM YOUR TESTERS ABOUT THE PROBLEM AND WHAT YOU'RE DOING TO FIX IT.**

Final Thoughts

Getting your app ready for external testers isn't just about the build—it's about the entire ecosystem surrounding the beta. By preparing technical, visual, and operational assets before releasing your beta build, you'll ensure a smooth and productive beta experience for your testers and team.

Use this guide as your roadmap to confidence and success BEFORE you start beta testing.

Why Centercode

Beta testing can be messy, but getting your app to testers doesn't have to be. Centercode simplifies the chaos with smart tools and integrations that **save time** and **make your life easier**.

Our **App Store Integration** with TestFlight and Google Play handles invites, access, and updates automatically, so you can skip the repetitive tasks and focus on what matters. Plus, first-class **Jira** and **Slack integrations** keep your team in sync and your feedback loop running smoothly.

At the heart of Centercode is **Ted AI**, a suite of powerful agents that take the guesswork out of beta testing. Ted simplifies workflows, automates repetitive tasks, and ensures actionable insights reach your team quickly. Features like **intelligent test plans**, **tailored surveys**, and **AI-enhanced reports** keep your beta on track, so you can focus on delivering a polished product. Whether it's seamless handoffs or ensuring your team gets the right insights at the right time, Centercode helps you deliver every release with confidence.

Ready to see the difference? Schedule a demo today!

Appendix

Template Resources

iOS - TestFlight

How to Install TestFlight and Join a Beta Test

iOS users, follow these steps to get started testing the beta version of our mobile app with TestFlight:

Step 1: Install TestFlight

TestFlight is required to install and test beta apps on your iPhone or iPad.

1. Open the **App Store** on your iOS device.
2. Search for "**TestFlight**."
3. Tap **Get** (or **Install**) to download and install the app.
4. Once installed, open **TestFlight** to ensure it's ready to use.

Step 2: Accept Your Beta Invitation

You should receive a beta invitation via email or a public link. Make sure to open the email or link on the mobile device where you'll be testing the app.

If you received an email invitation:

1. Open the email from [COMPANY NAME OR ACCOUNT NAME] on your mobile device.
2. Tap "**View in TestFlight**."
3. When TestFlight opens, tap **Accept**.

If you received a public link:

1. Click the **beta invitation link** on your mobile device.
2. When TestFlight opens, tap **Accept**.

Step 3: Install the Beta App

After accepting the invitation, you'll see the app listed in TestFlight:

1. Tap **Install** to download the beta version.
2. Once installed, open the app and start testing!

Android - Google Play Store

How to Join a Beta Test on Google Play Store

Android users, follow these steps to get started testing the beta version of our mobile app:

Step 1: Receive the Beta Invitation

You should receive a beta invitation via email or a direct link. Ensure you access this invitation on the **Android device** where you'll be testing the app.

If you received an email invitation:

1. Open the email from [COMPANY NAME OR ACCOUNT NAME] on your Android device.
2. Look for instructions or a link to join the beta program.

If you received a direct link:

3. Click the **beta invitation link** on your Android device.

Step 2: Join the Beta Program

The link will direct you to the [APP NAME] app's beta testing page in the Google Play Store:

1. Tap the "**Join**" or "**Become a Tester**" button.
2. A confirmation message will appear, indicating you've successfully joined the beta program.

Step 3: Install or Update the Beta App

1. After joining the beta program, navigate to the app's page in the **Google Play Store**.
2. If you haven't installed it yet, tap "**Install**" to download the beta version.
3. If the app is already installed, an update to the beta version will be available. Tap "**Update**" to install it.

centercode

Centercode is the leading platform for managing impactful in-the-wild user tests.

START FOR FREE, SCALE AS YOUR PROGRAM GROWS.

PLATFORM OVERVIEW



Run **real-world user tests** with passionate audiences to release amazing products every time.

PRODUCT LAUNCH

Engage your **Target Market** to successfully bring new products and services to market

USE CASES:

- Beta Tests
- User Acceptance Tests
- Field Tests

PRODUCT GROWTH

Recruit **Existing Customers** to continuously mature live products and services

USE CASES:

- Progressive Delivery
- Early Access
- Customer Advisory Boards

CUSTOMER ZERO

Gather **Employee** feedback to enhance your products while fostering a product culture

USE CASES:

- Alpha Tests
- Dogfooding
- Employee Tests

COMPETITIVE EDGE

Enlist **Competitors' Customers** to rapidly gain actionable intel about their products

USE CASES:

- Competitive Intelligence
- Market Testing
- Comparative Research

GET STARTED FOR FREE OR SCHEDULE A LIVE DEMO TO LEARN MORE

SCHEDULE A DEMO

START FOR FREE!

Why Centercode?

Increase Revenue

½ ★ **INCREASE**

in product ratings

Release Sooner

50% FASTER

test cycles

Gain Participation

75% INCREASE

in tester engagement

Save Time

40% + DECREASE

in time managing tests