Pre-release checklist for vibe-coded apps

A runbook for checking the exceptions in permissions, data, external calls, and deployment

A practical IXC resource for adoption decisions and delivery

Published 2026.09 · English edition · PDF + Excel worksheets

Using this resource

Sets out the expected behavior of an app built with generative tools and records the exceptions to check before release and the follow-up actions.

Developers who built a service through vibe coding, and product owners taking over the output of generative tools

Contents

01 Match implemented behavior to the intended rules 3

02 Check account and input boundaries 4

03 Check failures and repeats in external calls 5

04 Check data and deployment settings 6

05 Turn findings into fixes and revalidation 7

How to complete and reuse it

Read the criteria in this PDF, then fill the blank fields with your actual conditions and anything still unconfirmed. The Excel file for the same practice area has a worksheet for each resource. Keep the worked examples separate from real customer cases and confirmed quotes.

With attribution, you may quote and adapt this material for internal review and your own work. The latest files are at www.ixc.co.kr/en/assets.

Match implemented behavior to the intended rules

A screen that opens normally does not by itself confirm the product's rules. First, write down the expected result of each core flow. Separate the current behavior identified from screens and code, the behavior the product owner intended, and the policies not yet decided.

Requirement reconstruction sheet

Core flow	Intended rule	Current behavior and its evidence	Undecided policy and owner
Worked example: using a paid feature	Only accounts that have paid can use it	Record the access result for an ordinary account	Confirm the free trial scope

What to prepare	Purpose	What to check
Two test accounts and an admin role	Check owner and role boundaries	Each account's permissions and data
Test orders, files, and content	Check state changes and deletion	How to reset and clean up
Candidate build and change list	Pin down what the results apply to	Version and deployment environment
Test environments for external calls	Check failure and duplicate paths	Test keys and usage limits

Run the checks in a test environment and on test data that you administer. Do not mark an item PASS while its expected result is undecided; leave it for policy confirmation. Confirm one happy path first, then extend to the failure conditions of the same flow.

Check account and input boundaries

Treat permissions as distinct from hiding menus on screen. Check whether the server refuses the request when the URL of another account's data is accessed. For input, look at what gets saved with empty values, boundary values, and duplicate requests.

Flow to check	Condition to run	Expected result
Data ownership	Access account A's test data as B	Reading and changing limited by permission
Admin functions	Open admin paths from an ordinary account	Permission checked on the server
Session end	Repeat an earlier request after logging out	Actions that need authentication refused
Input boundaries	Empty values, negatives, and values above the maximum	Error message and saved state agree
Duplicate submission	Submit the same application or purchase repeatedly	No unintended duplicates created

Permission and input verification record

Flow and condition	Expected and observed result	Status and evidence	Action owner

If there is file upload, separately check the allowed formats and sizes and the permissions to view files. Compare whether the on-screen error message and the result the server saved mean the same thing. Record missing permissions and amount or data errors with their user impact, and set the order of fixes.

Check failures and repeats in external calls

For features that rely on external services, such as model calls, SMS, maps, and payments, check the delay, failure, and retry paths as well as success. Record how the number of requests and the data change when a user refreshes or presses the same button again.

Check item	Condition to run	Evidence to keep
Retry	Repeat the same request after a failure	Call count and final state
Response delay	No response within the agreed time	Waiting notice, stop, and resume behavior
Usage limit	Per-account limit reached	Server-side limit and on-screen message
Error information	External API returns an error	User-facing message and operations log

External call and cost checklist

Integration and flow	Limit and retry rules	Run result and evidence	Owner and action
Worked example: answer generation	Handle duplicate submissions and set a retry cap	Record the call count for the same request	Development owner to review

If the app has LLM features, also verify whether it takes input text as a command and uses it to access data or perform external actions. Look separately at tool execution permissions, downstream use of the output, and the scope of information passed outside. Record any path that executes a model response unchanged as a separate scenario.

Check data and deployment settings

Run flows that passed in the development environment again in the release candidate environment. Authentication domains and callbacks, payment environments, and missing environment variables produce different results even when the code is the same. Record setting names and purposes, but do not copy the secret values themselves into the result sheet.

Data and deployment checklist

Check item	Expected state	Evidence and result	Owner
Access to account and uploaded data	Permissions fit the owner and role		
Data sent externally	Only required fields sent		
Deletion and account-closure flows	Processing state matches product policy		
Environment variables, authentication domains	Match the candidate environment's settings		
Payment and webhook environments	Test and live settings kept separate		
Recovery and operations logs	Can restore the previous version and trace errors		

> **Worked example**
>
> If the login callback URL is still set to the development domain, rerun the login flow in the candidate environment. Record the version of the build with the corrected setting, what changed, and the URL actually reached. Also recheck the other login methods connected to the changed setting.

Confirm the recovery method by the result of running it, not by the existence of a command. If the data structure changes, also review whether it works with the previous app version. Record separately the owner who confirms the personal data handling policy and the owner who verifies the actual data flows.

Turn findings into fixes and revalidation

Which user flows are still unconfirmed decides the next task more than how many items passed. Classify FAIL as a problem to fix, BLOCKED as a problem in preparing the run, and NOT RUN as an item not yet run. Do not just change the status; attach the conditions needed for the next check.

Pre-release action list

Item and status	User impact	Owner and due date	Revalidation scope and evidence
Worked example: viewing another account's file / FAIL	Data exposed between accounts	Owner assigned to fix the permission check	Recheck direct access, shared links, and lists

Status found	Next action	Result to confirm
Expected result still undecided	Agree on the product rule	A scenario ready for a verdict
Core flow fails	Fix the cause and review the impact	Fix verification and regression on adjacent flows
Environment blocks the run	Prepare accounts, integrations, and data	Rerun under the same conditions
Rules missing in several areas	Reconstruct requirements and design the verification scope	List of core flows and exceptions

The review record keeps the build and scope, unresolved items, and the conditions to meet before release. IXC's AI-Built App Testing brings together requirement reconstruction, exception flow verification, and the regression scope a fix touches. Your completed requirement reconstruction sheet is where a consultation starts.

Publication and use

Based on your current build, your requirement reconstruction sheet, and the flows that failed or could not be confirmed, we propose the pre-release verification scope.

IXC Corp.

CEO Nam Kichul | Business registration no. 109-86-56732
4F 408, Chuncheon ICT Venture Center
7 Huseok-ro 462beon-gil, Chuncheon-si, Gangwon-do, Republic of Korea
+82-33-242-0210 | info@ixc.co.kr

www.ixc.co.kr

www.ixc.co.kr/en/services/ai-app-testing

www.ixc.co.kr/en/assets/ai-app-release-checklist

References

IXC AI-Built App Testing

https://www.ixc.co.kr/en/services/ai-app-testing

OWASP Authorization Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

OWASP Top 10 for Large Language Model Applications

https://owasp.org/www-project-top-10-for-large-language-model-applications/