Deciding between in-house push and a messaging tool

A decision kit that compares in-house operation, a tool, and running both against your current sending work and maintenance cost

A practical IXC resource for adoption decisions and delivery

Published 2026.09 · English edition · PDF + Excel worksheets

Using this resource

Decides whether to keep in-house app push or adopt OneSignal, based on your current work, the engineering burden, and the conditions for a switch.

Backend leads for push sending, CRM owners, and product leaders

Contents

01 Start by classifying the messages you send today 3

02 Count engineering time and operational waits together 4

03 Compare in-house operation, a tool, and running both 5

04 Verify the switch with one campaign 6

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.

Start by classifying the messages you send today

Some work is well suited to an in-house implementation. Look at who decides each message's audience, timing, and content, and the work a tool could reduce becomes clear.

If the recipient and the send time are fixed by business processing on the server, as with transaction status alerts, consider keeping your current direct sending. By contrast, if requests for audience selection, copy changes, scheduling, and experiments keep recurring, an operations tool becomes more valuable. Do not decide on send volume alone; look into the people involved and the waiting time from request to execution.

Send type	Who decides now	Question to check first
Transaction status alerts	Business system and server	Who handles failures and duplicates?
Service notices	Service operators	Do they change audience and timing directly?
Marketing campaigns	Marketing and engineering	Does each audience change need engineering?
Behavior-based messages	Event and journey owners	Is exit automatic after conversion?

Worked example

Order status alerts run reliably from the server, but marketing audience selection is tied to the engineering schedule. Consider keeping transactional sends in place while moving the marketing path to a tool. Set responsibility for consent and duplicate control across both paths at the same time.

Count engineering time and operational waits together

Beyond the first build, the cost of direct sending includes managing registration data, handling errors, keeping up with policy changes, and processing campaign requests. Compare using your current operating records.

Maintenance task	What to check in-house	What to check after adopting a tool
Device registration and refresh	App–server sync and cleanup	SDK switch and device linking
Failures and retries	Failure classification and duplicate prevention	Processing results and retry boundaries
Users and consent	Account switching and exclusions	Linking identifiers and consent
Campaign changes	Audience selection and waiting for deploys	Data intake and operating permissions
Performance measurement	Linking sends, clicks, and conversions	Metric definitions and analytics integration

The sending and registration specifications of FCM and APNs affect app and server maintenance. They need to be managed against the latest SDKs and platform documentation. Even after adopting a tool, the app's permission requests, account switching, deep links, and business event implementation remain. Only when the quote separates what the tool will do from what internal engineering will do can you see which scope will generate engineering requests again after adoption.

> **How to record the cost of waiting**
>
> For recent send requests, write down the date received, the date executed, any cancellation reason, and the people involved. Separate the waiting a tool reduces from the waiting that remains because of approvals or missing data. Do not count every delay as a saving from the tool.

Compare in-house operation, a tool, and running both

Compare how your current work gets done, and the evidence for it, rather than which features each option has. If failure handling for critical alerts is missing, fill that gap in the switch scope first.

Push operating model decision worksheet

Decision question	Current evidence	Option and our answer	Verification task, owner
What gets delayed while waiting on engineering?	Request, execution, and cancellation records		
Can the tool implement the audience conditions?	Test with data and a representative campaign		
Can subscriptions and consent be moved?	Review of the existing SDK and registration data		
After adoption, who operates what?	Table of roles, handover, and access		
How does the total cost change?	License, engineering, and operations quote		

Split the options into keeping in-house operation, adopting a tool, and running both by send type. For each option, write down the data, app updates, and operations owner it needs. Items whose behavior was not confirmed in testing stay open as further verification. Judge feasibility by actual audience selection, sending, and exit results rather than by check marks in a feature table.

> **Recording the verdict**
>
> If requests are few today and maintenance responsibility is clear, you can continue in-house operation. If the tool can handle the recurring requests and data conditions, consider adoption. Choose running both when you can explain how consent, exclusions, frequency, and metrics will be aligned across the two paths.

Verify the switch with one campaign

Rather than moving every send at the outset, check the data, subscriptions, and actual delivery of a representative campaign. Surface the parts that are hard to migrate before the contract.

- Confirm the existing SDK, subscription data by channel, consent, and who manages the sending credentials. Settle first how much can be migrated.
- Rewrite key campaigns as data conditions and entry and exit rules. Verify them with test accounts that represent real user conditions.
- When running both, set the send owner and the switchover time so that the two paths do not send duplicates to the same customer.
- Set the conditions for stopping if failures, metric discrepancies, or missing subscriptions appear. Retire the existing path only after comparing the actual results.

A successful send request in the new tool does not mean the migration is finished. Delivery to devices, link navigation, conversion measurement, and unsubscribe handling have to follow. Migration methods differ by channel and existing environment, so also check the app releases required and the impact on users. In the handover materials, record not only the settings but also the conditions tested and their results.

> **Scope to review with IXC**
>
> IXC provides OneSignal adoption, SDK and channel integration, data design, campaign rebuilds, and switch verification. Based on your current send request records and representative campaigns, we set with you which work stays in-house and the scope of adoption.

Publication and use

From your send types, engineering request history, SDK setup, and key campaigns, we determine the scope for in-house operation, tool adoption, and running both.

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/crm-migration

www.ixc.co.kr/en/assets/push-tooling-decision-guide

References

Firebase: FCM registration management

https://firebase.google.com/docs/cloud-messaging/manage-tokens?hl=en

Apple: User Notifications

https://developer.apple.com/Documentation/usernotifications/

OneSignal: Users

https://documentation.onesignal.com/docs/en/users

IXC CRM Migration & Transition

https://www.ixc.co.kr/en/services/crm-migration