PRODUCT CONCEPT / ANDROID

REMINDERS

Product Requirements Document

An independent concept for attaching a scheduled reminder to a WhatsApp message and keeping that message accessible in a dedicated inbox.

VERSION 1.0 | 16 NOVEMBER 2026

Prepared by Jahaan Desai

STATUS

Proposed product concept. This document defines a portfolio concept, not a shipped feature or an implementation commissioned by WhatsApp or Meta.

V1 platform scope: Android only

Document information

Field	Value
Document	Reminders Product Requirements Document
Document ID	RMD-PRD-001
Version	1.0
Status	Reviewed baseline for an independent product concept
Date	16 November 2026
Platform scope	Android only for V1
Product owner	Independent portfolio project by Jahaan Desai

> **SCOPE NOTE**
> This PRD describes a proposed Android experience. It is not evidence that WhatsApp or Meta has approved, built or shipped the feature.

Contents

- 1. Product Overview
- 2. Problem Statement and User Context
- 3. Product Goals, Non-Goals and Principles
- 4. User Stories and Core Use Cases
- 5. Functional Requirements
- 6. Edge Cases and System Behaviour
- 7. Non-Functional Requirements
- 8. Analytics and Success Metrics
- 9. Rollout and Release Plan
- 10. Risks, Dependencies and Open Questions
- Appendix A. Terminology and Event Notes
- Appendix B. Research Sources

1. Product Overview

> **TL;DR**
> Reminders lets a user attach one scheduled date and time to an individual WhatsApp message. When the reminder becomes due, the user receives a notification if settings allow it. The message remains in a dedicated cross-chat inbox after the notification is dismissed or delivered, so the user can return to the original conversation, reschedule or delete the reminder.

1.1 Product summary

Reminders is an independent product concept for WhatsApp. It connects two parts of the experience: a reminder action available from an individual message, and a persistent inbox that keeps every active reminder accessible across chats.

The inbox is the key part of the concept. A notification can bring a message back to attention at the selected time, but the reminder should not disappear just because the user dismisses the notification or is unable to act immediately. The inbox keeps the original message context available until the user deletes the reminder.

This is an independent portfolio concept, not an official WhatsApp or Meta feature. It does not claim to have invented message reminders or speculate about why any earlier experiment was changed or removed.

1.2 Product hypothesis

> **HYPOTHESIS**
> If users can schedule a reminder directly on a WhatsApp message and later find it in a persistent inbox, they may be more likely to return to messages they intended to act on, without relying on memory or transferring the task to another place.

This is a hypothesis to test. The research in this project establishes an initial problem signal within a narrow sample; it does not prove that this solution will improve follow-through or achieve broad adoption.

1.3 V1 scope

- Platform: Android only. Other platforms are outside this PRD.
- Creation: long-press one eligible message and choose the Reminders action from the message-selection toolbar.
- Scheduling: select a custom date and time using a single three-roller picker for date, hour and minute.
- Capacity: one active reminder per source message. Creating another reminder on the same message replaces the existing schedule.
- Access: a persistent Reminders shortcut in the WhatsApp Chats home-screen header opens the inbox.
- Persistence: the reminder remains in the inbox after its notification is delivered, dismissed or unavailable.
- Actions: open the source message, reschedule or delete the reminder.

1.4 What the feature is not

- A general-purpose bookmark, read-it-later or saved-items inbox.
- A shared task manager for groups or a system for assigning work to people.
- A replacement for a full task-management product.
- A completion tracker. V1 has no Done state, snooze action or completed-reminder history.

1.5 Proposed user and context

The initial audience is people who use WhatsApp heavily and regularly receive messages they intend to act on later. The exploratory research sample consisted of 36 people aged 21 to 28 from the project owner’s own network. This is a starting segment for discovery, not a claim that the problem is limited to this age group.

2. Problem Statement and User Context

TL;DR

I ran into this problem while juggling an internship, college, E-Cell responsibilities and an internship search. Messages often needed action later, but I could not always deal with them immediately. My workarounds helped me store or flag messages, but still depended on me remembering to revisit them. A small exploratory survey suggested that this was a familiar behaviour within my own network, although it cannot establish prevalence across WhatsApp users.

2.1 Personal trigger

While I was interning at Kotak811, handling college work and E-Cell responsibilities, and looking for my next internship opportunity, I was receiving messages that needed a reply or some action at a later time. I could not always deal with them when they arrived.

For time-sensitive items, I sometimes moved the task into a separate reminder or task app. For other messages, I took screenshots, starred or pinned them, forwarded them to myself, left them unread or relied on memory. Each workaround retained something, but none consistently brought the original message back to my attention at a useful moment.

2.2 Problem definition

PROBLEM STATEMENT

People receive WhatsApp messages they intend to respond to or act on later, but they cannot always deal with them immediately. Existing workarounds can preserve or flag the message, yet still rely on the user remembering to revisit it. The gap is not simply where to store a message; it is how to bring that specific message back to the user's attention at the intended time.

2.3 Exploratory survey

I shared a short questionnaire with 36 friends and colleagues aged 21 to 28 who use WhatsApp extensively. The results are self-reported and directional. The sample is a convenience sample, not a representative sample of WhatsApp users.

Finding	Responses	Interpretation
Receive messages that need later action several times a week	27 of 36 (75.0%)	The trigger situation was frequent in this sample.
Forgot or significantly delayed at least one follow-up in the past 30 days	33 of 36 (91.7%)	Self-reported follow-up friction was common in this sample.
Usually leave a message unread when unable to act immediately	29 of 36 (80.6%)	Unread status is a common workaround in this sample.
Selected distraction from other messages or tasks as a difficulty	21 of 36 (58.3%)	Distraction was the most frequently selected difficulty.
Have ever used a separate app for a task originating in WhatsApp	13 of 36 (36.1%)	This is a different survey question from the usual-workaround question.

Counter-signal: 6 of 36 respondents (16.7%) selected that they generally did not have difficulty following up. This should remain visible in the interpretation rather than being omitted.

2.4 What the survey does and does not tell me

- It suggests that messages requiring later action and forgotten or delayed follow-ups are familiar within this small group of WhatsApp-heavy users.
- It shows that leaving messages unread and relying on memory were more commonly selected than transferring every item to another task app.
- It does not show that every WhatsApp user has this problem, that the behaviour is more common among younger users, or that users would adopt Reminders.
- It does not measure real product usage, task completion or the effect of a reminder on behaviour.

2.5 Other qualitative and secondary signals

Three informal conversations were useful in shaping the direction. I recall that all three people used an external app to track tasks that originated in WhatsApp. These were informal conversations, not structured interviews: I do not have recordings, formal notes or verbatim quotations. They should be described as directional input, not formal qualitative research.

A public Reddit discussion from August 2025 includes users reporting that they had seen and used a WhatsApp message-reminder option before it disappeared from their apps. The saved capture showed 28 votes and 32 comments at the time it was archived. This is a self-selected qualitative signal, not evidence of market-wide demand, retention or the reason for the feature's disappearance.

A separate Medium case study by Jatin Jakhar explores a related WhatsApp reminders idea and includes its own small survey. Its findings belong to that author and are not combined with my 36 responses.

2.6 Evidence, assumptions and opportunity

Statement	Evidence status	How it is used
I found it difficult to return to particular messages	Personal experience	Starting point for discovery.
33 of 36 respondents forgot or delayed at least one follow-up	Self-reported convenience sample	Directional problem signal only.
People may value a persistent inbox after the notification fires	Product rationale and hypothesis	A design choice to test, not a validated finding.
Reminders may improve message follow-through	Unvalidated hypothesis	Measured in a future pilot.

The opportunity I chose to explore is a way to defer action on a specific message without requiring the user to transfer it elsewhere or remember to look for it again.

3. Product Goals, Non-Goals and Principles

> **TL;DR**
>
> V1 is focused on scheduled reminders attached to individual messages. It should retain the original conversational context, keep reminders visible after notifications fire and let users decide when an item is no longer needed. It should not become a general task manager or infer task completion.

3.1 Product goals

- Make it possible to schedule a reminder directly from an eligible WhatsApp message.
- Keep the reminder visible in one cross-chat inbox after the notification has been delivered, dismissed or suppressed.
- Preserve a live reference to the original message and its conversation context, subject to normal message access and expiry rules.
- Make overdue reminders easy to identify and reschedule.
- Give users control over when a reminder is removed, without implying that the associated task is complete.
- Measure whether users return to the original message and whether the experience is reliable enough to justify wider release.

3.2 Non-goals for V1

- Shared group task lists, assignment, ownership or permission management.
- A general-purpose saved-items or read-it-later system.
- Multiple active reminders on one source message.
- Bulk reminder creation from multiple selected messages.
- Snooze, Done, completed-task history or automatic task completion.
- Preset scheduling shortcuts such as Later today or Tomorrow.
- An overdue-only inbox or automatic overdue-first sorting.
- Preserving disappearing or view-once message content beyond its original WhatsApp lifecycle.

3.3 Product principles

Principle	Product implication
Keep the original context	A reminder points back to the message; it is not a disconnected text note.
Do not let a dismissed notification erase intent	The inbox retains the reminder until the user deletes it.
Keep the user in control	No automatic completion state. Deletion means the reminder is no longer wanted, not necessarily that the task is done.
Avoid unnecessary task states	No Done, snooze or completed history in V1.
Use familiar WhatsApp interactions	Use long press, the message-selection toolbar and an established home-screen entry point.
Make overdue status actionable	The overdue alarm indicator opens the rescheduling picker directly.
Respect existing privacy and message rules	The feature must not bypass chat locks, view-once restrictions or disappearing-message expiry.

4. User Stories and Core Use Cases

> **TL;DR**
> The main journey is: select a message, schedule a time, find the reminder in the inbox, return to the source when it becomes due, and reschedule or delete it when needed.

4.1 Primary user context

The primary V1 user is a frequent WhatsApp user who receives messages that require a later reply or action and wants to defer that action without losing the message context. The initial research group was aged 21 to 28, but the concept is not restricted to that demographic.

4.2 User stories

ID	User story	Priority
US-01	As a user, I want to set a reminder on an individual message so I can return to it at a chosen time.	Must
US-02	As a user, I want to see all active reminders in one inbox so I do not have to search across chats.	Must
US-03	As a user, I want a due reminder to notify me when settings allow it.	Must
US-04	As a user, I want a reminder to remain in the inbox after the notification so I can act later.	Must
US-05	As a user, I want to open the original message from its reminder so I retain the conversation context.	Must
US-06	As a user, I want to reschedule an overdue reminder directly from the inbox.	Must
US-07	As a user, I want to delete a reminder without deleting the source message.	Must
US-08	As a user, I want reminders to reflect message edits or unavailable-source states accurately.	Must
US-09	As a user, I want reminders to remain consistent across linked devices.	Should

4.3 Core use cases

UC-01 Create a reminder

1. The user long-presses one eligible message in a chat.
2. The selection toolbar shows the alarm action immediately to the left of Reply.
3. The user taps the action and the single-screen date, hour and minute roller picker opens.
4. The user chooses a future date and time, then confirms.
5. The reminder is saved and appears in the Reminders inbox.

UC-02 Open the Reminders inbox from the home screen

The user taps the persistent Reminders shortcut in the Chats home-screen header. The inbox opens directly. This shortcut is an inbox entry point, not a create-reminder action.

UC-03 Open a reminder from a notification

The user taps a due notification. WhatsApp opens the Reminders inbox, navigates to the relevant entry and highlights it briefly. The user then opens the original message from that entry.

UC-04 Open the source message

The user taps a reminder card outside the Read more control, or taps its source arrow. WhatsApp opens the original conversation at the source message. Returning restores the previous inbox scroll position.

UC-05 Reschedule an overdue reminder

The user taps the alarm indicator beside an overdue reminder. The same date and time picker opens directly. Saving updates the existing reminder, clears its overdue state and moves it to the top in Date added sorting.

UC-06 Delete a reminder

The user deletes one reminder or selects multiple reminders for bulk deletion. Only the reminder entries are removed. Bulk deletion provides a five-second Undo action.

UC-07 Handle an unavailable source

If the source message is no longer accessible, the reminder remains with an appropriate placeholder or explanation. Source navigation is disabled when the original message cannot be opened.

5. Functional Requirements

PRIORITY KEY

Must means required for the proposed V1 experience. Should means important, but subject to technical validation or release sequencing. Open means a decision is not yet settled and is listed again in Section 10.

5.1 Message selection and reminder creation

ID	Requirement	Priority
FR-01	The user must be able to start reminder creation from an individual eligible message using the existing long-press interaction.	Must
FR-02	The Android message-selection toolbar must show an alarm action immediately to the left of Reply when exactly one eligible message is selected.	Must
FR-03	The alarm action must be hidden when multiple messages are selected. Bulk reminder creation is not supported in V1.	Must
FR-04	Tapping the alarm action must open one picker screen containing date, hour and minute rollers.	Must
FR-05	The picker must initially show the device's current local date and time. The selected values must be visually distinct from surrounding values.	Must
FR-06	Dates before the current local date must be disabled. Times earlier than the current time must be disabled when Today is selected.	Must
FR-07	The user must be able to choose a future time today or a time on a future date. No preset time options are included in V1.	Must
FR-08	If the selected time is less than 60 seconds away, the reminder must fire immediately after confirmation. If the time passes while the picker is open, it must fire immediately after confirmation.	Must
FR-09	The user must confirm the selected schedule before it is saved. Closing or cancelling the picker must not create a reminder.	Must
FR-10	Only one active reminder may exist per source message. Starting creation again on a message with an active reminder must replace its schedule without creating a duplicate inbox entry.	Must

5.2 Notifications and scheduling

ID	Requirement	Priority
FR-11	A reminder must become due at its scheduled local date and time. The reminder due state must be tracked separately from whether a system notification is posted.	Must
FR-12	When notification settings permit, a system notification must be posted at the due time, including when the user is currently in the source chat.	Must
FR-13	Multiple reminders due in the same minute must generate separate notifications. V1 does not group them.	Must
FR-14	Reminder notifications must respect device and WhatsApp notification settings, including the user's mute preferences.	Must
FR-15	Reminders originating in archived chats must not generate reminder notifications. They must still become overdue and remain in the inbox.	Must
FR-16	If system notifications are disabled, reminders must still be saved, become overdue and remain visible in the inbox. An in-app notice must appear only inside the Reminders inbox according to the frequency rule in Section 6.	Must
FR-17	If the device is powered off at the due time, the reminder must be marked overdue after restart. A delayed notification must not be sent on restart.	Must
FR-18	Reminder state must sync across linked devices. Notifications may be posted on linked devices subject to each device's settings; exact deduplication and ownership rules require technical validation.	Should

5.3 Reminders inbox and navigation

ID	Requirement	Priority
FR-19	The Chats home-screen header must include a persistent Reminders shortcut in this left-to-right order: Payments, Reminders, Camera. Reminders must sit immediately to the left of Camera.	Must
FR-20	If Payments is unavailable, the Reminders shortcut must retain its position immediately to the left of Camera.	Must
FR-21	Tapping the Reminders shortcut must open the Reminders inbox directly. It must not initiate reminder creation.	Must
FR-22	The inbox must show active reminders from all chats and retain each entry after notification delivery, dismissal or suppression until the user deletes it.	Must
FR-23	Each entry must show enough source context to identify the conversation and message, plus the scheduled reminder date and time. Long message previews must support inline Read more expansion.	Must
FR-24	Tapping the card outside Read more, or tapping its source arrow, must open the original conversation at the source message. Returning must restore the previous inbox scroll position.	Must
FR-25	The inbox header count must show the total number of active reminders. It must not combine overdue and total counts.	Must
FR-26	The inbox must support sorting by Date added or Remind me. Date added is the default; newest creation or rescheduling appears first. Remind me sorts by scheduled time, soonest first.	Must
FR-27	change Date added order.	Must
FR-28	Search must support loose matching across message content, sender name and source conversation name. Exact indexed fields require engineering confirmation.	Must

5.4 Overdue state, badge and reminder actions

ID	Requirement	Priority
FR-29	When the scheduled time passes, the reminder must remain in the inbox and be marked overdue. The original scheduled date and time must remain visible.	Must
FR-30	A small alarm indicator must appear beside overdue reminders only. It must not appear beside reminders that are not overdue.	Must
FR-31	clear overdue status and update Date added order.	Must
FR-32	The Reminders shortcut must show a red badge counting overdue reminders that have not yet been seen in the inbox. Opening the inbox marks all currently overdue reminders as seen and clears their contribution to the badge.	Must
FR-33	A reminder that becomes overdue after the inbox has been viewed must be eligible to contribute a new unseen-overdue count. Seen status must not imply completion or deletion.	Must
FR-34	Deleting an individual reminder must remove only the reminder, not the original message. There is no Done action or completed-reminder history in V1.	Must
FR-35	Bulk deletion must support multi-select for deletion only and show an Undo action for five seconds. Swipe-to-undo is not supported.	Must

5.5 Source-message lifecycle

ID	Requirement	Priority
FR-36	If the source message is edited, the inbox preview must reflect the latest content and show an Edited label.	Must
FR-37	If the sender deletes a message for everyone, retain the reminder and show the sender-deleted placeholder. Source navigation remains available if the chat is accessible.	Must
FR-38	If the user deletes a message for themselves, retain the reminder, show an unavailable-source explanation and disable source navigation.	Must
FR-39	If the user deletes their own message for everyone, retain the reminder and show the You deleted this message placeholder. Source navigation remains available to that placeholder.	Must
FR-40	If a chat is cleared, deleted or left, retain the reminder and disable source navigation when the source is no longer accessible.	Must
FR-41	Reminders may be created on messages in disappearing-message chats. If the source expires before the reminder is dealt with, retain the reminder but replace the content with an expiry placeholder. Do not preserve expired content.	Must
FR-42	Reminders may be created on view-once messages, but must never restore or preserve access to view-once content after its original lifecycle ends.	Must
FR-43	Reminders must not be created for messages in locked chats. The exact disabled or error state remains an implementation detail to specify.	Must
FR-44	Blocking the sender alone must not disable source navigation if the source message remains accessible.	Must

6. Edge Cases and System Behaviour

TL;DR

The reminder record and notification state are separate. A reminder can be upcoming, due or overdue while its notification may be delivered, suppressed or missed. The inbox is the persistent source of truth for active reminders; message access rules continue to apply.

6.1 Reminder lifecycle

State	Entry condition	Expected behaviour
Upcoming	A reminder is saved and its scheduled time has not arrived.	Visible in the inbox; no overdue indicator.
Due	The scheduled local date and time is reached.	The reminder becomes due. A notification is posted only if settings and chat rules permit it.
Overdue and unseen	The scheduled time has passed and the user has not opened the inbox since it became overdue.	Visible in the inbox with an alarm indicator; contributes to the unseen-overdue badge.
Overdue and seen	The user has opened the inbox after the reminder became overdue.	Remains overdue with its alarm indicator. No longer contributes to the unseen-overdue badge.
Rescheduled	The user changes the schedule of an existing reminder.	The same entry receives the new schedule, clears overdue status and moves to the top in Date added sorting.
Deleted	The user deletes the reminder.	The reminder entry and future reminder action are removed. The source message is not deleted.

6.2 Notification state is separate

A reminder's lifecycle must not be inferred from notification status. For example, an overdue reminder may have had its notification delivered and dismissed, may have had notifications suppressed by settings, or may have been missed because the device was off. In each case, the reminder itself remains available in the inbox until deleted.

Reminder state must synchronise across linked devices. The current concept expects due notifications on linked devices, subject to each device's notification settings. The implementation must prevent duplicate records and reconcile simultaneous edits or deletes. Notification ownership, delivery ordering and conflict resolution require technical validation before release.

Condition	Notification behaviour	Reminder behaviour
Notifications enabled and chat eligible	Post at due time.	Remain in inbox; become overdue if not rescheduled or deleted.
Muted chat	Respect the applicable mute preference.	Remain in inbox and become overdue normally.
Archived chat	Do not post reminder notification.	Remain in inbox, become overdue and may contribute to unseen badge.
System notifications disabled	Do not post system notification. Show the in-app notice according to the frequency rule.	Remain in inbox and become overdue.
Device off at due time	Do not send a late notification after restart.	Mark overdue after restart and retain in inbox.
Multiple reminders due in the same minute	Post separate notifications where allowed.	Keep separate reminder entries.

6.3 Notification-disabled warning frequency

- Show the first in-app warning the first time the user opens the Reminders inbox while system notifications are disabled.
- Show one additional warning only if reminder creation in the latest seven days is more than 15% higher than in the preceding seven days, notifications remain disabled and a minimum activity floor is met.
- After the second warning, do not show another warning.
- The 15% threshold is provisional and not evidence-based. The minimum activity floor is still open. The notice appears only inside the Reminders inbox, not on the Chats home screen.

6.4 Date and time picker behaviour

- Use the device's current local date and time when the picker opens.
- Use one screen with three rollers: date, hour and minute. The selected values sit in a central highlight band; surrounding values are visually de-emphasised.
- Dates before Today are disabled. When Today is selected, times earlier than the current local time are disabled.
- A future date may use any valid time. A same-day reminder may use a later time on the current day.
- If the chosen time is less than 60 seconds away, fire immediately after confirmation. If it passes while the picker is open, fire immediately after confirmation.
- The same picker is used for changing an existing schedule, including from the overdue alarm indicator.

6.5 Source-message and conversation changes

Source event	Reminder entry	Source navigation
Message edited	Show the updated message and Edited label.	Available.
Sender deletes for everyone	Show the sender-deleted placeholder.	Available if the chat remains accessible.
User deletes for everyone	Show You deleted this message.	Available to the placeholder.
User deletes for me	Retain reminder with an unavailable-source explanation.	Disabled.
Chat cleared or deleted	Retain reminder with a source-unavailable explanation.	Disabled if the message is no longer accessible.
User leaves group	Retain reminder with a source-unavailable explanation.	Disabled if the conversation is no longer accessible.
Disappearing message expires	Retain reminder, replace content with an expiry placeholder and do not preserve expired content.	Unavailable for the expired message.
View-once lifecycle ends	Retain only an entry consistent with the original view-once restrictions. Do not restore content.	No access to expired view-once content.
Sender is blocked	Retain reminder.	Available if the source remains accessible.

6.6 Locked chats

A reminder cannot be created from a locked chat. The UI should explain why the action is unavailable without exposing locked-chat content. The exact disabled state and copy are open for design and implementation.

6.7 Deletion and Undo

- Individual deletion removes only the reminder, not the source message. Whether single-item deletion also offers Undo remains open.
- Bulk deletion supports selecting multiple reminders. A confirmation snackbar indicates the number deleted and offers Undo for five seconds. Swipe-to-undo is not supported.

7. Non-Functional Requirements

TL;DR

The experience depends on reliability and trust. Scheduling, notifications, source references and linked-device state must behave consistently. Performance and privacy should meet WhatsApp's existing Android product standards; implementation targets must be set with engineering before release.

Area	Requirement	Validation approach
Performance	The selection action, picker, inbox and search should feel responsive and align with existing WhatsApp Android performance baselines.	Measure p50 and p95 latency on supported low, mid and high-range devices. Set numeric release targets with engineering.
Scheduling reliability	A saved reminder must not be silently lost, duplicated or scheduled at an unintended time.	Test scheduling, app process termination, device restart and repeated user actions.
Notification reliability	Due notifications must follow the saved schedule and applicable notification settings.	Test enabled, disabled, muted, archived, device-off and linked-device scenarios.
Data consistency	Only one active reminder may exist per source message. Updates and deletes must propagate across linked devices.	Test simultaneous edits, schedule replacement, deletion and reconnect behaviour.
Source integrity	The inbox must reflect the current source-message state without bypassing message expiry, view-once or chat-lock restrictions.	Test every source lifecycle case from Section 6.
Privacy and security	Use existing WhatsApp access controls. Do not expose message content from locked or unavailable chats. Analytics must not record message text or personal conversation details.	Security review, permission tests and analytics payload inspection.
Accessibility	Controls must work with Android accessibility services, have clear labels and states, and retain adequate contrast and touch target sizes.	TalkBack, font scaling, contrast and touch-target testing.
Battery and resource use	Scheduling should not depend on frequent background polling or unnecessary wake-ups.	Battery and background activity profiling across representative devices.
Localization	Date, time, number and language presentation should follow WhatsApp's Android localization conventions.	Locale, 12/24-hour clock and language testing.

7.1 Release quality bar

- No unresolved critical or high-severity defects in reminder creation, scheduling, notification behaviour, source-message handling, data consistency or privacy.
- No known issue that can create duplicate active reminders for the same source message.
- Analytics events used for the pilot must be verified before users enter the pilot.
- The product must preserve normal message access and expiry rules in all tested states.

8. Analytics and Success Metrics

> **TL;DR**
> The primary outcome is Reminder Follow-through Rate: the share of eligible reminders after which the user opens the original source message within 24 hours of the scheduled time. This is a proxy for returning to the message, not proof that the task was completed. Proposed decision thresholds are below 20%, 20% to below 40%, and 40% or higher; these are hypotheses for a pilot, not industry benchmarks.

8.1 Measurement approach

The measurement plan separates the primary user outcome from adoption, engagement and reliability. A reminder being created is not sufficient evidence that the feature helps, and a source-message open is not evidence that the user completed the task.

8.2 Primary metric: Reminder Follow-through Rate

> **DEFINITION**
> Reminder Follow-through Rate = eligible reminder instances that lead to an open of the original source message within 24 hours after the scheduled due time, divided by all eligible reminder instances with a complete 24-hour observation window.

For measurement, an eligible reminder instance is an active reminder that reaches its scheduled due time and whose source message is accessible at that point. Reminders deleted or replaced before their scheduled due time are not counted as due instances. A reminder that is rescheduled creates a new scheduled instance for measurement at its updated due time. The numerator counts a source-message open from the reminder experience; merely opening the inbox or tapping a notification does not count. If the source becomes unavailable during the 24-hour observation window, exclude that instance from the primary denominator and report it separately as a source-availability issue.

This metric is a behavioural proxy for returning to the message. It cannot establish whether the user replied, bought something, completed a task or otherwise followed through outside WhatsApp.

8.3 Supporting metrics

Dimension	Metric examples	What it helps answer
Discovery and adoption	Eligible exposed users who view the Reminders action; users who create at least one reminder.	Can users find the feature, and do they try it?
Repeat usage	Users creating reminders in more than one week; reminders per active reminder user.	Does the feature become useful beyond first use?
Creation friction	Picker opens, completed creations, cancelled picker sessions and schedule replacements.	Where do users abandon or change the flow?
Reminder engagement	Notification opens, inbox opens, source opens and reschedules.	What do users do when reminders become due?
Inbox health	Overdue reminders per active user, overdue age and deletion patterns.	Are reminders helping users return, or accumulating without action?

8.4 Guardrail metrics

- Notification reliability, including scheduling failures, missed notifications where delivery should have been permitted, and duplicate alerts.
- Unexpected or duplicate active reminder records for the same source message.

- Rapid deletion or replacement after creation, which may signal accidental activation or poor scheduling controls.
- Notification opt-outs, complaint rates and other signals of notification fatigue.
- App stability, performance, battery use and linked-device consistency.
- Source-message access and expiry violations. Any privacy or access-control violation is a release blocker.

8.5 T1 and T2 decision thresholds

Observed Reminder Follow-through Rate	Interpretation	Proposed product response
Below 20%	Below T1. The current approach is not demonstrating enough follow-through.	Do not expand. Investigate discovery, timing, notification and inbox friction. If the result remains low after targeted improvements and adequate evidence, stop the current approach.
20% to below 40%	Between T1 and T2. The experience may be useful but needs improvement.	Keep the core experience stable, identify the largest friction points and test focused changes.
40% or higher	At or above T2. A promising signal that users return to source messages.	Consider gradual expansion and reducing steps in the creation flow, provided guardrails remain healthy.

The 20% and 40% thresholds are proposed hypotheses, not external benchmarks and not findings from the 36-person survey. A release decision requires a sufficient number of eligible reminder instances, a complete observation window and healthy guardrails. The minimum event volume is still open.

8.6 Instrumentation events

Event	When it fires	Useful properties
reminder_action_viewed	The user sees the Reminders action for an eligible single-message selection.	entry_point, eligibility_state
reminder_picker_opened	The scheduling picker opens.	entry_point, create_or_edit
reminder_created	A new reminder is saved for a message with no active reminder.	entry_point, schedule_lead_time_bucket
reminder_rescheduled	An existing reminder receives a different schedule, including replacement from the message action.	entry_point, change_reason, previous_state
reminder_due	The saved schedule reaches its due time.	notification_eligible, chat_state, device_state
reminder_notification_posted	The system notification is posted.	device_type, notification_settings_state
reminder_notification_opened	The user opens a reminder notification.	entry_point, device_type
reminders_inbox_opened	The user opens the Reminders inbox.	entry_point, active_count_bucket, unseen_overdue_count_bucket
reminder_source_opened	The user opens the original source message from a reminder.	entry_point, reminder_state
reminder_deleted	The user deletes an individual or selected reminder.	delete_mode, reminder_state
reminder_bulk_delete_undone	The user uses Undo after bulk deletion.	deleted_count_bucket
reminder_source_unavailable	The source can no longer be opened from its reminder.	reason_category

Do not emit a separate reminder_replaced event. A replacement from the message action emits one reminder_rescheduled event with entry_point=message_context_action and change_reason=replace_existing. Analytics must not log message text, media, contact names, phone numbers or group names; use pseudonymous identifiers and aggregated reporting with appropriate access and retention controls.

9. Rollout and Release Plan

> **TL;DR**
>
> The proposed rollout has four named phases: pre-release validation, a limited Android pilot, pilot evaluation and gradual expansion. The stages are deliberately defined without inventing cohort sizes or rollout percentages that have not been established.

9.1 Phase 1: Pre-release validation

Before exposing the feature to users, validate the core experience across supported Android configurations. This phase is about reliability, permission handling and correct message lifecycle behaviour.

- Create, replace, reschedule and delete reminders.
- Validate the three-roller picker, past-date and past-time restrictions, near-immediate reminders and confirmation behaviour.
- Test notifications for enabled, disabled, muted and archived chat states, including multiple reminders due in the same minute.
- Test device restart, linked-device sync, duplicate prevention and concurrent changes.
- Test edits, deletions, disappearing messages, view-once messages, locked chats and unavailable conversations.
- Verify analytics event names, properties, privacy and metric calculations before the pilot.

> **EXIT CRITERIA**
>
> No unresolved critical or high-severity defects in reminder creation, scheduling, notification behaviour, source-message handling, data consistency or privacy.

9.2 Phase 2: Limited Android pilot

Make the feature available to a limited group of Android users. The proposed pilot duration is four weeks so that the evaluation can observe repeated use rather than only first-time interactions. The specific cohort size and access mechanism remain implementation decisions.

- Observe whether users discover the Reminders action and create reminders successfully.
- Monitor whether due notifications are posted as expected and whether reminder records remain consistent.
- Measure whether users return to the source message within 24 hours of the scheduled time.
- Review rescheduling, deletion, overdue accumulation, complaints and notification opt-outs.
- Segment results by notification eligibility and chat state where relevant.

9.3 Phase 3: Pilot evaluation and release decision

At the end of the pilot, assess the results against T1 and T2 in Section 8. Interpret the thresholds only when there is enough eligible activity and the observation window is complete. A promising primary metric does not override a failure in privacy, notification reliability or stability guardrails.

9.4 Phase 4: Gradual expansion

If the pilot meets the outcome and reliability criteria, expand access to additional Android users in stages. Review the same primary metric and guardrails after each expansion. Pause the expansion if reliability, privacy or user-impact signals deteriorate. Before release, define how already-created reminders and scheduled notifications behave during a rollback; existing reminders should not be discarded unexpectedly.

10. Risks, Dependencies and Open Questions

> **TL;DR**
>
> The concept and core user-facing behaviour are defined, but several implementation details remain open. The most important are linked-device notification rules, time and timezone changes, the minimum activity floor for a second in-app notice, pilot sample size and a few prototype-to-spec alignment details.

10.1 Key risks and mitigations

Risk	Why it matters	Proposed mitigation
Missed, delayed or duplicate notifications	Users may stop trusting reminders if alerts do not arrive when expected or arrive more than once.	Validate scheduling and notification behaviour across supported Android configurations. Track failures and duplicate alerts.
Follow-through proxy mistaken for completion	Opening a message does not mean the associated task was completed.	Describe the metric as a source-message return proxy, not task completion.
Original message becomes unavailable	The reminder may remain active after the user can no longer open its source.	Retain the reminder with a specific placeholder or explanation and disable navigation where appropriate.
Notifications are suppressed	Muted, archived or notification-disabled states may prevent an alert.	Retain the reminder in the inbox, expose overdue status and apply the unseen-overdue badge rules.
Linked-device inconsistency	Devices may show different reminder state or generate duplicate notifications.	Validate sync, idempotency, event ordering and notification ownership.
Low feature discovery	Users may not notice the shortcut or understand how to create a reminder.	Measure discovery and creation funnel events during the pilot; use observed friction to guide improvements.
Privacy or message-lifecycle violation	A reminder must not bypass locked-chat, view-once or disappearing-message rules.	Preserve WhatsApp access controls and validate every source lifecycle scenario before release.

10.2 Dependencies

- Android scheduling and notification capabilities that can reliably support saved reminder times and device restart behaviour.
- A stable message reference that reflects edits, deletions, expiry and conversation availability without bypassing normal access restrictions.
- Persistent reminder storage and synchronisation across linked devices.
- Integration with WhatsApp and Android notification settings, including mute and archive behaviour.
- Privacy-conscious event instrumentation and reliable calculation of the primary metric.
- A controlled rollout mechanism with the ability to pause further access and define rollback handling.

10.3 Open questions before production release

Open question	Why it needs resolution
What minimum activity floor should apply to the second notification-disabled warning?	A 15% increase can be misleading when the number of reminders created is very small.
What happens when the user changes timezone, device clock or daylight-saving settings?	The product needs a consistent rule for whether a reminder follows the original instant or the new local clock.
How should linked devices decide which notifications to post?	The concept expects sync and device-level notifications, but ownership, deduplication and race conditions need definition.
What minimum pilot sample or event volume is required to evaluate T1 and T2?	A four-week pilot may not generate enough eligible reminders for a meaningful decision.
What is the rollback behaviour for existing reminders?	Users should not unexpectedly lose saved reminders or receive confusing notifications during a rollback.
What is the exact search index?	The intended fields include message content, sender name and source conversation name, but technical indexing rules are not specified.
What is the final locked-chat disabled state and copy?	The restriction is decided, but its user-facing explanation has not been specified.
Which unavailable-source explanation should each scenario use?	Delete for me, clearing a chat, deleting a chat and leaving a group have different causes and should not be conflated.
Should single-item deletion offer Undo?	The five-second Undo is defined for bulk deletion; the single-item interaction is not finalised.
How is the notification-disabled warning state synchronised?	The first and second warning limits must not reset unexpectedly across linked devices.

10.4 Final implementation note

These open questions are not hidden assumptions. They are deliberately carried into the implementation backlog. A production team would close them through engineering validation, design review and pilot evidence before broad release. The approved prototype uses Date added as the sort label, and this PRD follows the same terminology. The sort places the most recently created or rescheduled reminder first; opening the source does not change its position. This remains a reviewed product concept baseline, not a claim that every technical detail is production-ready.

Appendix A. Terminology and Event Notes

Term	Meaning in this PRD
Reminder	An active schedule attached to one source message.
Source message	The WhatsApp message from which the reminder was created.
Reminders inbox	The persistent cross-chat view of active reminder entries.
Due	The scheduled local date and time has been reached.
Overdue	The scheduled time has passed and the reminder has not been rescheduled or deleted.
Unseen overdue	An overdue reminder that has not been included in an inbox visit since becoming overdue.
Date added sort	Newest reminder creation or rescheduling first.
Remind me sort	Order by scheduled reminder date and time.

A.1 Analytics event clarification

The event reminder_replaced is not emitted as a separate event. When a user starts the message-level creation flow on a message that already has an active reminder, the existing schedule changes. Record one reminder_rescheduled event with entry_point=message_context_action and change_reason=replace_existing. This preserves the distinction between a new reminder and a change to an existing reminder without counting one action twice.

A.2 Sorting label clarification

Date added is the default sort. It places the most recently created or rescheduled reminder first. Opening the source message does not change its position. Remind me sorts by scheduled date and time, soonest first by default. The approved prototype and this PRD use the same labels.

Appendix B. Research Sources

The sources below informed discovery or product context. They are not evidence that this proposed product experience has been validated.

Jatin Jakhar, "From Chatting to Organizing: WhatsApp's Reminders": https://medium.com/design-bootcamp/from-chatting-to-organizing-whatsapps-reminders-998069adf3de

Reddit discussion, "Remind me function disappeared" (18 August 2025):
https://www.reddit.com/r/whatsapp/comments/1mt865n/remind_me_function_disappeared/

WABetaInfo, WhatsApp beta for Android 2.25.21.14 (22 July 2025): https://wabetainfo.com/whatsapp-beta-for-android-2-25-21-14-whats-new/

Black and McBride, "Effects of delay and reminders on time-based prospective memory in a naturalistic task" (2024):
https://pubmed.ncbi.nlm.nih.gov/38968418/

Li, Zhang and Ao, "Why is instant messaging not instant?" (2023):
https://www.sciencedirect.com/science/article/pii/S0747563223000067

Microsoft Research and CHIIR, "Exploring Email Triage: Challenges and Opportunities" (2019):
https://www.microsoft.com/en-us/research/wp-content/uploads/2019/02/Email_Triage_CHIIR19.pdf

Research handling notes:

- Jahaan's 36-response questionnaire is the only survey represented as primary research in this PRD.
- Jatin's separate survey must not be combined with the 36 responses.
- The Reddit thread is a qualitative, self-selected signal. Its saved vote and comment counts are archival snapshot values.
- Public reports about different WhatsApp reminder experiments should not be conflated. No claim is made about why any experiment disappeared.