

# The Morning Brief

February 2026 | Author: [Your Name] | Status: Draft

## 1. The Problem

Every morning, tens of millions of knowledge workers perform the same unconscious ritual: open Slack, scan email, glance at calendar, check Jira. They're not planning their day. They're reacting to whatever showed up overnight. By the time they start "real work," they've already surrendered their attention to whichever app yelled loudest.

This happens because every SaaS app creates its own attention queue—and each queue is independent, chronological, and blind to the others. Your actual priority list doesn't exist in any single app. It's scattered across Slack @mentions, Jira tickets, PR reviews, Doc comments, email threads, and calendar invites. The only way to assemble it is in your head, manually, every morning.

**The deeper issue is that there is no planning layer for the workday.** Every app helps you communicate, track, or create—but no app helps you step back and answer: "What should I spend my time on today?" Without that, people stay stuck in receiver mode: responding to inputs instead of making deliberate choices. Important items slip through not because people are careless, but because they have no system that gives them the full picture.

## 2. Our Approach

Many products have tried to become the starting point for the workday and failed—unified inboxes tried to replace existing tools; dashboards tried to become where you live all day. We are building something different: a **Morning Brief**—an AI-generated, personalized daily briefing delivered as a **conversation**, not a dashboard. The agent walks you through everything needing your attention, one item at a time, and helps you handle it without ever leaving. Three principles:

- **Insert before, don't replace.** A 10–15 minute conversation before you open Slack or email. You still use all existing tools—you just start the day with full awareness.
- **Self-contained.** Users never leave the brief. Quick tasks are handled inline by the AI assistant; everything else is noted on a shared scratchpad. The moment a user leaves to "quickly check Slack," they don't return.
- **Complete, then confident.** The goal is reaching the end of the conversation. Every item seen, every decision made—that moment is the emotional payoff that brings users back tomorrow.

### 3. Why Glean

- **Cross-app data:** 100+ real-time connectors index Slack, email, Jira, Docs, GitHub, Calendar with permission-aware access. The brief's value is proportional to its completeness—no competitor has this breadth.
- **Enterprise + Personal Graph:** Org structure, team ownership, individual work patterns. Enables prioritization contextual to each user.
- **Action layer:** Execute actions inside connected apps (reply in Slack, update Jira, accept invites). Makes the brief self-contained.
- **Browser extension:** New-tab takeover intercepts the user's first browser action of the day—a natural delivery surface.

### 4. Target Persona

**Engineering Managers and Product Managers** at companies already using Glean. Both have high cross-app attention demand, sit at dependency intersections (when they miss something, others are stuck), and default to receiver mode.

#### Key User Value

**Confidence:** “I’ve seen everything that needs my attention today.”

**Clarity:** “I know what’s most important and I have a scratchpad for my day.”

**Control:** “I’m starting my day proactively instead of reacting to whatever is loudest.”

### 5. Core Requirements

- **Conversational, sequential experience.** The brief is a conversation, not a page. The agent presents one item at a time. The user responds, and the next item appears. No list to scan and abandon. This prevents choice paralysis, ensures every item gets attention, and enables a future voice modality—every message is one spoken paragraph, every action is a simple utterance.
- **Completeness.** Surface every item needing attention today, across all connected apps. One missed item breaks trust permanently.
- **Trust through transparency.** After the main brief, the agent tells the user how many items it reviewed and filtered out, and offers to walk through what didn’t make the cut—conversationally, in batches. Users can “pull in” any filtered item, teaching the system to weight similar items higher.
- **Prioritization.** Rank by actual importance, not recency. Objective signals (blocking impact, deadlines, requester relationship) always override personal preference.

- **The Scratchpad.** A shared, living document between the user and the agent. The agent jots observations throughout the day as it watches apps. The user adds notes anytime. During the brief, every action writes to the scratchpad. It persists across sessions, feeds into tomorrow's brief, and over time becomes long-term memory.
- **Memory.** Four layers: long-term (30 days bootstrap), short-term (daily), org-level (Enterprise Graph), behavioral (brief interactions + scratchpad patterns).
- **Self-containment.** Every item processable without leaving. Free-text input alongside quick-action buttons—edit drafts, ask for context, write custom replies.

## 6. North Star Vision (12 Months)

### The Experience

It's 9 AM. You open your browser. Your Morning Brief starts a conversation: "19 items came through overnight. 8 need you, 3 are urgent. I've been jotting notes to your scratchpad." It shows your calendar with prep flags—sprint planning has 3 tickets at risk, your manager 1:1 shows a note you wrote yesterday: "bring up headcount."

You tap "Got it, what's urgent?" One item appears: Marcus and two engineers, blocked 3 days. You see who's blocked, the cascade, a draft reply. You send it. The agent notes "Decided: event sourcing" on your scratchpad and moves on.

Next: a follow-up from yesterday's 1:1, extracted from the transcript. By 9:12, every item is handled. The agent says: "That's everything. I also reviewed 11 other things. Want to see what didn't make the cut?" Your scratchpad is full: decisions, reminders, your notes. Tomorrow, the agent reads it to generate the next brief.

### The Scratchpad

The scratchpad replaces the traditional "plan." It is a shared working surface, not a to-do list:

- **Agent writes:** Observations ("Marcus asked about architecture decision again, 4:32 PM"). Decisions during brief ("Approved Dev's PR"). Reminders ("API rate limiting due Friday"). Carryover from previous briefs.
- **User writes:** Notes anytime ("Bring up headcount in 1:1"). Context the agent doesn't have ("Loop in CTO"). The agent reads these and connects them to meetings and brief items.
- **Persistence:** Carries across sessions. Yesterday's unresolved items appear in today's brief. Patterns feed into long-term memory over time.

### The Memory System

**Long-term memory:** 30 days of historical data bootstraps before the first brief. Key relationships, active projects, response patterns, weekly rhythm.

**Short-term memory:** Updated daily. Meeting action items, active threads, scratchpad carryover.

**Org-level memory:** Enterprise Graph surfaces team sprint health, cross-team blockers, incident propagation.

**Behavioral memory:** Every brief interaction is a signal. “Send reply” = trust AI drafts. “Skip” = deprioritize. Editing a draft teaches voice. “Tell me more” reveals missing context. Pulling in filtered items corrects thresholds.

**Scratchpad memory:** Accumulated from agent observations and user notes over time. Recurring themes, user-noted priorities, and carryover rates all feed back into long-term memory weekly, shaping what the brief surfaces and how it’s ordered.

## How Memory Generates the Brief

Memory is not a background feature—it is the engine that produces the brief. Every morning, the system reads all five memory layers to make three decisions for each candidate item:

- **Should it appear?** Long-term memory determines relevance (is this your project? your collaborator?). Org memory determines organizational importance (does this affect your team?). Behavioral memory applies the user’s personal threshold (do they engage with items like this?). Objective importance can override behavioral signals—a blocking PR surfaces even if the user usually ignores GitHub.
- **Where should it rank?** Short-term memory provides recency and urgency (thread from yesterday, deadline tomorrow). Scratchpad memory reveals what the user is actively thinking about (noted items get elevated). Long-term memory provides relationship weight (messages from top collaborators rank higher).
- **How should it be presented?** Behavioral memory shapes the experience: users who edit drafts see longer, more detailed drafts. Users who approve PRs quickly see shorter summaries. Users who ask “tell me more” on certain item types automatically get expanded context next time.

**Transparency:** Every item in the brief shows a “Why this is here” explanation that explicitly cites which memory layers contributed to the decision. This makes the system’s reasoning visible and correctable. The full memory profile is viewable and editable by the user at any time.

## The Memory Loop

Memory is not static. It improves through a continuous feedback loop: Memory generates brief → user triages items → triage actions become behavioral memory → scratchpad notes

accumulate → scratchpad patterns feed long-term memory → next brief is more personalized. By day 30, the brief feels like it was written by a chief of staff who's worked with you for a year.

## Voice Modality

The conversational architecture supports voice as a primary modality. Every agent message is one spoken paragraph. Every user action is a simple utterance. Switching from buttons to voice is a modality swap, not a redesign.

## 7. MVP (3 Engineers, 3 Months)

**Core hypothesis:** Will people adopt a conversational morning brief as the starting point for their workday, if it walks them through everything needing their attention—one item at a time—and lets them act on it without leaving?

**What we ship:** Conversational brief (calendar + one-at-a-time items + filtered review). Scratchpad with agent observations and user notes. Memory from 30-day bootstrap + daily updates + brief interactions. Five connectors: Slack, Calendar, Jira, GitHub, Meeting Transcripts. Triage: act on it, remind me later, skip, plus free text. Browser extension + Slack DM delivery.

**What we defer:** Voice modality (architecture ready, ship text-first). Inferred deadlines from conversations. Delegate with person-suggestion. Gmail and Google Docs connectors.

## 8. Key Tradeoffs

- **Conversation vs. dashboard.** One item at a time prevents cherry-picking but requires trust. The “what didn’t make the cut” mechanism provides the escape valve.
- **Scratchpad vs. structured plan.** Freeform shared surface, not a to-do list. Less structure in exchange for richer signal and natural collaboration between user and agent.
- **AI prioritization vs. user rules.** Lead with AI. Triage actions and scratchpad patterns serve as implicit correction.
- **Self-contained vs. deep-linking.** Every action resolves inline. Non-negotiable: leaving = reactive mode.
- **Memory from day 0.** Bootstrap from 30 days of data. Generic first experience = notification aggregator.

## 9. Validation Approach

**Phase 1 — Concierge (weeks 1–2):** Human-compiled briefs for 5–10 EMs/PMs via Slack DM. Did it surface anything missed? Would they want it tomorrow?

**Phase 2 — Internal dogfood (weeks 1–4):** Glean's own EMs/PMs use early builds. Validates conversational flow and prioritization.

**Phase 3 — Closed beta (months 2–3):** 10–20 users at 2–3 customers. Measures success metrics.

## 10. Success Metrics

### North Star: Brief Completion Rate

% of workdays where the user reaches the end of the conversation. Target: >50% within 3 months.

Metric	Definition	Target
Daily Active Usage	% completing brief before 10 AM	>40% DAU/MAU
Week-4 Retention	% of week-1 users still active	>60%
Scratchpad Engagement	% adding $\geq 1$ note/day outside brief	>20%
Missed Item Rate	User-reported items not surfaced	<10%
Filter Override Rate	% of filtered items pulled in	<15%

## 11. Risks

**Trust requires completeness.** One missed item breaks trust. Mitigation: five connectors, “what didn’t make the cut” transparency, Missed Item Rate metric.

**Users break out of the conversation.** Sequential reveal requires patience. Mitigation: self-contained actions, short items, visible progress.

**Scratchpad becomes noise.** Agent writes too much and value drops. Mitigation: observations and decisions only, not raw notifications. Signal density threshold.

**Objective vs. personal conflict.** Blocking PR surfaces even if user ignores GitHub. Mitigation: objective importance always overrides.