

## PORTFOLIO INTRODUCTION

# Sanjay Shrestha

## Senior Product Designer

15 years designing digital products, from a government portal reaching 4 million citizens to enterprise SaaS used by 100,000+ professionals. Founding designer at Decisions — I build the research, systems, and UX practice behind the product, not just the screens.

**15+**

Years of experience

**4M+**

Citizens reached

**100K+**

Professionals served (Decisions)



# What's Inside

A quick map of what's ahead. Jump to any section.

00

## Experience

15 years across government, SaaS, and enterprise platforms.

P. 03

01

## Decisions AI — Mobile App

Rethinking a desktop-first Teams platform for native mobile.

P. 04

02

## Dashboard Redesign

One role-aware dashboard for organizers, attendees & guests.

P. 17

03

## AI Recap & Minutes

Turning meeting chaos into automatic decisions & action items.

P. 30

04

## Let's Connect

Contact details and next steps.

P. 40

## EXPERIENCE

# 15 Years Building Digital Products.

From founding designer to senior product leader - across government platforms, enterprise SaaS, and B2B collaboration tools.

**2019 – PRESENT**

### Lead Product Designer

Decisions

Founding designer - built the UX function and design system from scratch, now serving 100K+ users across 2,500+ companies.

**2016 – 2019**

### Product Designer

Webscale Networks

Sole designer for the Mojo Stratus SaaS platform - redesign cut support tickets 40% and handoff errors 35%.

**2014 – 2016**

### UX Consultant

Microsoft

Designed the Singapore CPF portal for 4M+ citizens with full WCAG 2.0 accessibility compliance.

**2009 – 2014**

### UI-UX Designer & Front-End Developer

Early Career · Kathmandu / Bangalore

Delivered UX across agencies and enterprise clients; introduced QA standards and mentored junior designers.

# 01

## Decisions AI

### From Desktop-First to Mobile-First

*I helped transform a desktop-first Microsoft 365 meeting platform into a focused native mobile experience.*

A native iOS and Android app with 50,000+ installs on Google Play alone, replacing a Teams web view that took 5 to 10 seconds to load.

Decisions AS · 2020 – Present · 5.0★ App Store · 4.3★ Google Play

[View Design Appendix →](#)

*Original product, Information Architecture, Wireframes, Iterations, Role experiences*

## THE CHALLENGE

# We Had a Product. I Had to Rethink the Context.

Decisions is a secure meeting management platform built natively on Microsoft 365 and Teams, used across financial services, government, energy and professional services. Before this project, the mobile experience was essentially the web app resized for a smaller screen.

### EXISTING REALITY

- 50+ enterprise features
- Desktop-first workflows
- Responsive mobile web
- Built for Microsoft Teams

### MOBILE REALITY

- Quick, in-between-moments interactions
- Touch-first, not mouse-first
- Offline, camera and microphone access
- Push notifications

### DESIGN CHALLENGE

*How might we create a native mobile experience without trying to replicate the entire desktop application?*

# Before Designing, I Needed to Understand the Mobile Moment.

We ran 1:1 interviews with meeting participants across four groups. The goal wasn't to validate a feature list; it was to find where the current experience broke down and which mobile moments mattered most.

## WHO WE TALKED TO

Board members & executives, executive assistants, project managers, field & on-site team leads.

**What surprised me:** *I expected people to ask for more mobile features. Instead, they asked for faster access.*

### Board members & executives

*"I review pre-reads on the plane. I need the documents to work without wifi."*

### Meeting attendees

*"See my meeting agenda in 5 seconds and find out what I need to do after the meeting."*

### Executive assistants

*"I need to see the agenda clearly on my phone and know if anything's changed before I walk in."*

**Key insight:** speed, offline access, and discoverability, not missing features, were the real barriers.

# I started with mobile as a smaller web app. Research showed it needed to become a different tool.

*That reframing shaped my design direction: design for native context, not just adaptive layout.*

**DESIGN PRINCIPLE** Mobile shouldn't compete with desktop. It should complement it.

### Teams Add-on (Adaptive)

- Web UI resized for mobile
- No offline capability
- No device hardware access
- Feature-complete, but unusable on the move

### Native Mobile App

- Designed for touch from the start
- Offline-first for meeting books
- Agenda CRUD Operation
- Focused on the workflows research surfaced

# I Prioritized Around Mobile Value, Not Feature Parity.

*Every feature had to justify its place.*

50+ Desktop Features

Research Insights

Mobile Context

Business Value

Technical Feasibility

**Focused version 1**

### Why these made the cut

**Meeting Dashboard** → because users wanted to know what was happening in seconds, not minutes.

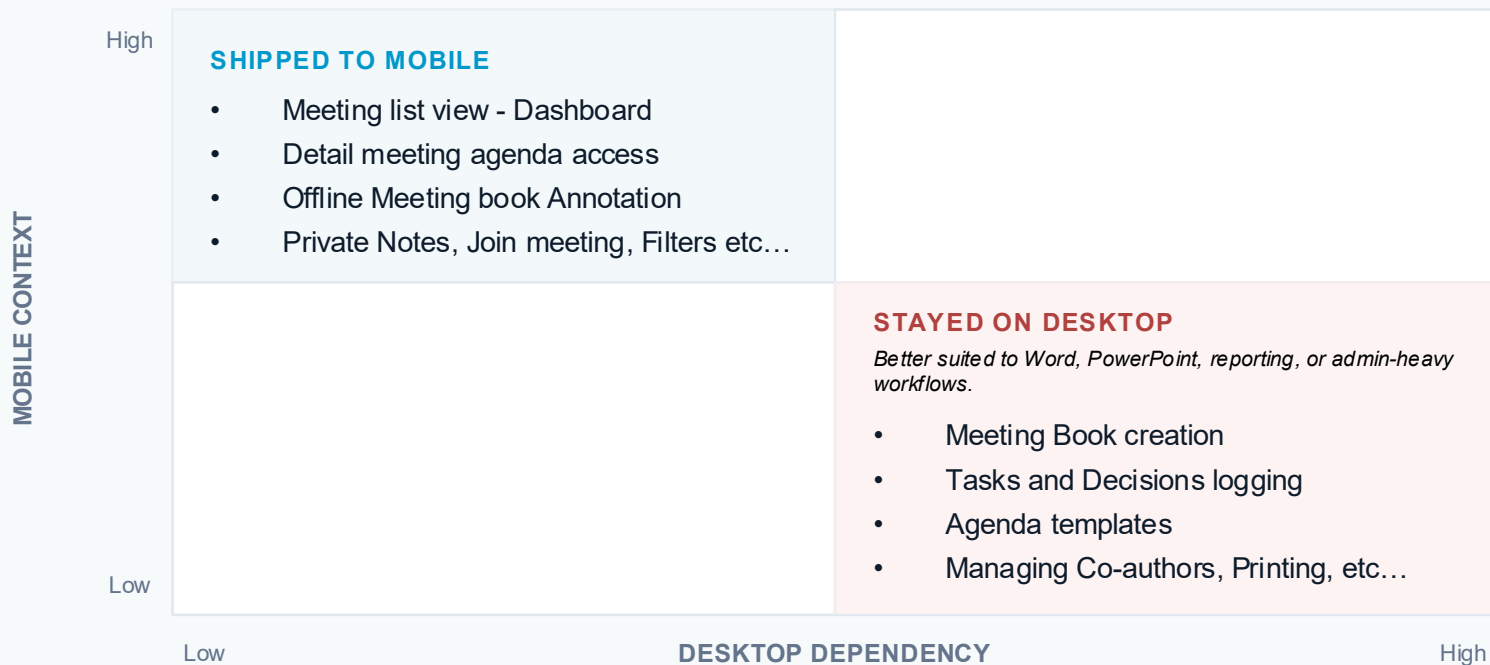
**Agenda viewing** → because attendees needed quick access to agenda items and attachments before meetings.

**Meeting Book (Offline)** → because executives frequently reviewed pre-reads while travelling or without internet.

**Comments** → because conversations were already happening in Teams and users shouldn't have to switch contexts.

# Not Every Desktop Feature Belonged on Mobile.

I prioritized features that had strong mobile value and low dependency on desktop workflows.



# What I Prioritized for the First Release.

Instead of recreating the desktop experience, I focused Version 1 on the moments users needed most: before, during, and between meetings.

| BEFORE THE MEETING                                      | DURING THE MEETING                                                       | ANYWHERE                                                      |
|---------------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>Meeting Dashboard</b><br>Today's meetings, briefly   | <b>Comments</b><br>Teams chat, right from the agenda                     | <b>Offline Access</b><br>Pre-downloaded books, no wifi needed |
| <b>Agenda Viewing</b><br>Review agenda details          | <b>Meeting Book Annotation</b><br>Highlight, draw and comment naturally. |                                                               |
| <b>Meeting Book</b><br>View/Annotate the compiled PDF   |                                                                          |                                                               |
| <b>Attachments</b><br>Open agenda attachments instantly |                                                                          |                                                               |

# Designed for How People Actually Use Their Phone in a Meeting.

Four screens carry the weight of the research. Each one started as a problem, not a layout.

### Meeting Dashboard

#### PROBLEM

People check meetings while walking between rooms.

#### DESIGN DECISION

Surface only what users need before the next meeting.

#### RESULT

**A 30-second check before walking in.**

### Meeting Book Reader & Annotation

#### PROBLEM

Executives were annotating printed PDFs by hand.

#### DESIGN DECISION

Native annotation, built to work offline.

#### RESULT

**No paper, notes stay attached to the file.**

### Agenda & Attachments

#### PROBLEM

People needed details without hunting through Teams.

#### DESIGN DECISION

Browse items and open files from the agenda.

#### RESULT

**Details on demand, in the flow.**

### Comments

#### PROBLEM

Meeting conversations were happening outside the app.

#### DESIGN DECISION

Integrate Teams conversations directly into the meeting flow.

#### RESULT

**Organizers and attendees stay in sync.**

# Building native gave us room to evolve.

Starting with a focused native foundation made it possible to introduce new capabilities without redesigning the experience.

1

## Offline Access

Executives review pre-reads on flights and in low-signal buildings. Pre-downloaded meeting books stay open with no connection.

2

## Annotation

Board members were marking up printed PDFs by hand. Native drawing and highlighting replaced that habit entirely.

3

## In-Person Transcription

The Teams add-on only worked inside a Teams call.

Native microphone access enabled in-person meeting capture, something the Teams add-on could never support.

4

## AI Recaps

People didn't want pages of transcript. They wanted to understand what they missed. AI Recaps transformed conversations into decisions, actions and summaries.

[View Design Appendix](#) →

*Original product, Information Architecture, Wireframes, Iterations, Role experiences*

# How the Mobile Foundation Evolved After V1.

After launch, I kept evaluating the mobile roadmap through the same lens: user feedback, product usage, and market movement. As AI notetaking tools raised expectations, the next opportunity became clear: mobile needed stronger capture and recap workflows.

1

## Focused launch

Version 1 deliberately prioritized the highest-value mobile workflows.

2

## Continued discovery

User feedback and market trends revealed new expectations around AI-assisted meetings.

3

## Expanded the opportunity

Native capabilities created room for in-person capture and richer meeting intelligence.

4

## Extended the platform

AI Notetaker and AI Assistant were added without rethinking the core mobile experience.

## Why the foundation mattered

### We designed for evolution

The first release solved today's problems without limiting tomorrow's opportunities.

### Native kept paying off

The mobile foundation supported AI capabilities without major redesign.

### Discovery didn't stop

Research continued after launch through customer feedback, usage patterns and competitive analysis.

*The biggest lesson wasn't that AI changed the product. It was that thoughtful prioritization made continuous innovation possible.*

# Designing What Comes Next

*Not parity. Matching each feature to the context it's used in.*

### DESKTOP-FIRST workflows

- Meeting Book creation
- Managing Co-author
- Meeting Minutes generation
- Digital signatures (Adobe Sign)
- Agenda templates creation
- More...

### MOBILE CANDIDATES to validate

- Agenda creation from scratch
- Cross-meeting search
- AI Agenda suggestions
- Meeting Minutes generation
- Auto Recap Generator
- More...

### MOBILE-NATIVE capabilities

- Offline access
- Meeting Book annotation
- In-person transcription (camera & mic)

# Success Wasn't Just Internal.

I cannot share detailed internal metrics, but public app store data, product stability, and shipped capabilities still show the direction of impact.



### Quality held up in public

- 5.0★ on the App Store,
- 4.3★ on Google Play.

One reviewer calls it “excellent for meeting management and decision tracking.”



### A new use case, unlocked

In-person transcription brought field and on-site teams onto Decisions for the first time, a segment the Teams add-on could never reach.



### Stability outpaced growth

0.63% crash rate and 0.06% ANR on Android, holding steady even as installs climbed past 50,000 on Google Play.



### Momentum, not a one-off

The public changelog shows releases every few weeks: smarter recaps, shareable recaps, Intune reliability fixes, each traceable to a real signal.

# What This Project Taught Me.

### Parity vs. clarity

I started this project believing mobile needed feature parity. Research taught me it needed clarity, not parity. That single shift influenced every decision we made afterward.

### The hardest feature

The hardest design decision was not what we built. It was what we deliberately chose not to build.

### What native really means

Native wasn't a technology decision. It was a commitment to designing for context instead of convenience.

### What I'd explore next

Making competitive monitoring a formal part of the backlog process, not an ad hoc trigger, so market shifts get caught as early as user feedback does.

*The best version of a product isn't the most complete one. It's the one that's honest about what it's for.*

# 02

# Dashboard Redesign

Reimagining the Decisions dashboard into a role-first meeting workspace for 500k+ users.

Decisions AS · 2019 – Present · Web, Responsive & Microsoft Teams · 4 Major Releases

[View Design Appendix →](#)

*Original product, Information Architecture, Wireframes, Iterations, Role experiences*

## THE CHALLENGE

# One Dashboard. Three Frustrated Users.

The Decisions dashboard tried to serve everyone. In practice, that meant it served no one well. Organizers needed quick prep and status visibility. Attendees needed agenda and action items briefly. Guests needed just enough to show up.

### Organizer

*"I click into each meeting to see what's needed. There's no way to tell immediately."*

### Attendee

*"I just want to see the agenda and what was decided. Where do I find that?"*

### Guest

*"I only need the meeting link and time. Everything else is noise."*

Organizer



Attendee



Guest

**All landed here.** One interface. Three expectations.

### Scope & Constraints:

- 50k+ organizers,
- 500k+ participants
- Web application, multiple browsers
- WCAG AA required
- Must support A/B testing architecture

*The question that shaped every design decision:*

## How could I design one dashboard that still felt purpose-built for different meeting roles?

### OLD ARCHITECTURE

One dashboard

Same experience

Everyone

→ **Problems**



### NEW PRINCIPLE

**Role-aware**

**Context-aware**

**Meeting-state aware**

# My initial hypothesis was that navigation was the problem.

My first assumption was that navigation was the problem. I believed users were struggling because they had to click too much to understand what needed attention.”



## OUR HYPOTHESIS GOING IN

*If we simplify the tab structure, users will find what they need.*

UI

### User Interviews

Interviewed organizers, attendees, and guests

- 8 interviews,
- one-on-one.

AA

### In-App Analytics

Which tabs were used? Where did users drop off?

CA

### Competitive Analysis

Multi-role UI patterns:



UA

### Usability Testing

5 rounds, from wireframes through final design.

# The Real Problem Wasn't Navigation. It Was Identity.

01

## Different Roles. Different Mental Models.

Organizers plan, attendees look back, and guests just need logistics: three different mental models, not three flavors of one.

02

## “What do I do next?” was the real question

Users were not browsing the dashboard. They were trying to understand their next action quickly.

03

## Guests were invisible in the design, and they noticed

Guests got the same full dashboard as an organizer, when all they needed was a time, a place, and a join link.

## REFRAMING THE PROBLEM

**I thought I was simplifying navigation. Research showed me I was solving the wrong layer of the problem.**

### Initial assumption

The list view was too click-heavy. Fewer steps to meeting detail would solve it.



### Discovery

Organizers, attendees, and guests have three different jobs to be done, each needing a different “first thing I see.”

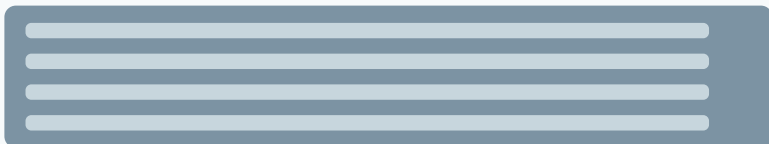


### Design shift

We moved from a feature-first list to a role-first information architecture.

# Starting Simple. Learning Fast.

## HYPOTHESIS #1 · V1 BASIC LIST VIEW



### GOAL

Reduce clicks — let organizers scan meetings fast.

### WHAT HAPPENED

Users scanned quickly, but couldn't tell which meetings needed action without clicking in. No status was visible at all.

### LEARNING

Fast scanning isn't enough. People need status without clicking.

## HYPOTHESIS #2 · V2 STATUS + ACTION CARDS



### GOAL

Give organizers status and next actions at a glance.

### WHAT HAPPENED

Organizers responded positively because they could understand meeting status without opening every meeting.

### LEARNING

Solving for one role can create a new gap for another.

*Each iteration was tested with real users before moving forward. We didn't guess, we confirmed.*

## Solving Cold Start. Then Discovering a New Problem.

V3 · TEMPLATE SYSTEM

**+23%**

agenda creation rate  
(template adoption)

*New organizers stopped freezing on a  
blank agenda.*

V2 helped organizers see status, but new organizers still froze with no agenda yet. V3 solved the cold-start problem for organizers, but testing showed the attendee journey was still too buried.”

- **What WORKED**

Template adoption climbed. New organizers stopped staring at a blank page.

- **What we learned**

The attendee experience was still an afterthought. The recap felt secondary on every screen. Testing showed attendees kept scrolling past prep tools, looking for post-meeting content buried too deep.

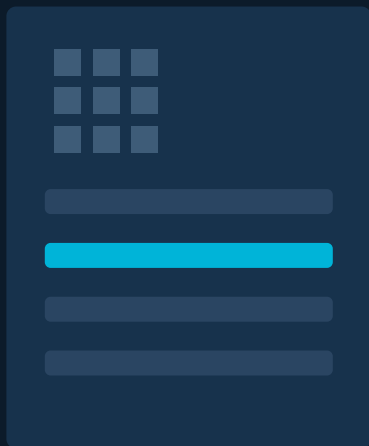
- **Key realization after v3**

Pre-meeting and post-meeting are two completely different jobs, and the dashboard needed to handle both, simultaneously, for different people.

## THE FINAL DIRECTION

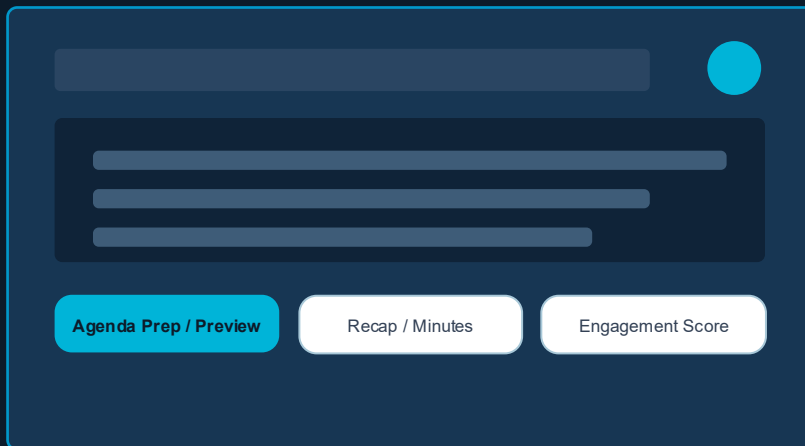
# V4: Unified, Recap-First, Role-Aware.

By V4, I had clarity on the core ingredients: status visibility, contextual actions, template access, and recap-first post-meeting content.



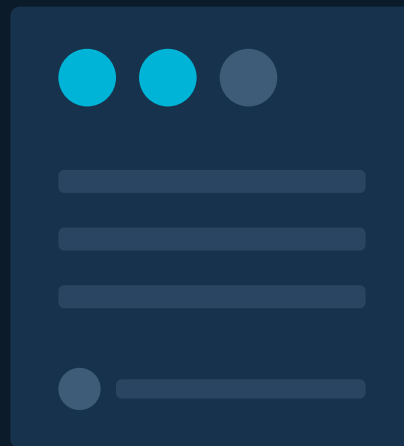
### 1 Navigation

Calendar picker + scrollable meeting list, always visible.



### 2 Context

Role-adaptive content: prep tools, AI Recap, or minimal logistics.



### 3 Collaboration

Comments, RSVP tracking, and activity log.

*Role-based rendering means organizers, attendees, and guests each see a version tailored to their job, without three separate dashboards to maintain.*

## FINAL SOLUTION

# Every Decision Traces Back to a Research Insight.

Each final design decision was tied directly to a research signal, not personal preference.



### Recap moved above the fold

**INTERVIEW** *"I couldn't find the summary."*

**OUTCOME** +31% engagement



### One toolbar, two roles

**ANALYTICS** *Attendees never clicked 'Create Agenda.'*

**OUTCOME** Simpler to build & maintain



### Calendar picker on left

**INTERVIEW** *"David (manager) tracked 10+ meetings/week."*

**OUTCOME** Faster for power users



### Settings hidden by default

**TESTING** *New users skipped setup when options appeared upfront.*

**OUTCOME** Simpler onboarding

# Same Layout. Three Different Experiences.

## Organizer

The Organizer role view is divided into two sections: PRE-MEETING and POST-MEETING. The PRE-MEETING section features an Engagement Score badge, a prominent blue 'Create Agenda' button, and a 'Generate Recap' button. The POST-MEETING section displays a large blue progress bar at the top, followed by a grey progress bar and a blue progress bar at the bottom.

- Engagement score badge, visible at a glance
- Create Agenda / Generate Recap front and center
- Post-meeting: Full AI Recap hero + Decisions & Action items

## Attendee

The Attendee role view is divided into two sections: PRE-MEETING and POST-MEETING. The PRE-MEETING section includes four buttons: 'View Agenda', 'Request Items', 'Suggest Topic', and 'My Notes'. The POST-MEETING section features a large blue progress bar at the top, followed by a grey progress bar and a blue progress bar at the bottom.

- Prep cards: View Agenda, Request Items, Suggest Topic
- Post-meeting: Full AI Recap above the fold
- Read-only agenda summary, no clutter

## Guest

The Guest role view is divided into two sections: PRE-MEETING and POST-MEETING. The PRE-MEETING section shows a grey progress bar and a prominent blue 'Join' button. The POST-MEETING section displays a grey progress bar.

- Just the meeting time and a Join button
- No collaboration tools, nothing extra to interpret
- Same minimal layout, before and after

## OUTCOMES & IMPACT

# The Numbers Followed the Design Decisions.

**+23%**

**Agenda creation rate**

*Suggested templates removed  
blank-page freeze*

**+31%**

**Recap engagement**

*Recap moved above the fold*

**-18%**

**Task completion time**

*Faster action-item processing*

**+42%**

**Collaboration rate**

*Cross-team signals surfaced*

**< 30s**

**Faster scanning**

*For guest to find meeting info*

*Based on internal product analytics.*

*V4 became the foundation for all subsequent dashboard features. The component library now serves 10+ engineers.*

## REFLECTION

# What Five Years on One Problem Taught Me.

### What surprised me most

The biggest lift didn't come from new features; it came from moving existing ones. The AI Recap existed before V4. Surfacing it above the fold for attendees drove a 31% engagement increase.

Sometimes the biggest UX improvement isn't building something new. It's putting the right thing in the right place.

### What I'd do differently

I'd push for guest persona research earlier. We treated guests as a simple edge case, but their fast-completion story turned out to be a strong differentiator. That deserved dedicated discovery, not a footnote.

### What I'd test next

A scoped recap for guest presenters: just the agenda item they presented, the decisions made, and any action items involving them. The hypothesis: guests who feel informed are more likely to follow through.

### The biggest lesson

Designing one adaptable system turned out to be stronger than designing three separate dashboards. The constraint of one dashboard, not three, forced a level of systems thinking that made the component library stronger than it would have been otherwise.

[View Design Appendix](#) →

*Original product, Information Architecture, Wireframes, Iterations, Role experiences*

# 03

# AI Recap & Minutes

From scattered, manual meeting notes to structured intelligence — automatically.

Decisions AS · 2023 – present · Web/Teams App + Native Mobile app · AI Feature

[View Design Appendix →](#)

*Original product, Information Architecture, Wireframes, Iterations, Role experiences*

# Every Meeting Left Work Behind.

Before AI, I saw a gap between the product promise and the actual user experience: Decisions helped structure meetings, but the outcome still depended on someone manually doing the admin work afterward.

The platform's core promise depended entirely on whether someone did the admin work after the meeting ended.

## Scope & Constraints

- Microsoft Teams + Azure OpenAI + Whisper
- Angular web app - M365 ecosystem (Planner, SharePoint)
- Design goal: zero extra work for users

### ● Meeting Owner

*"I spend 20 minutes after every meeting manually logging tasks and decisions. It kills momentum."*

### ● Participant

*"I can never find what was decided or who was responsible. I just ask someone again the next week."*

### ● Team Manager

*"I have no visibility into meeting outcomes unless someone emails me. There's no paper trail."*

Design question: *How do we make every meeting automatically produce the decisions, tasks, and summaries it was supposed to — without asking anyone to do extra work?*

# I Started by Testing the Assumption

My initial assumption was that the logging process was too slow. Research showed me the issue was deeper. The manual logging process was broken because it was tedious. We thought we just needed to reduce friction. 12 user interviews across 6 enterprise customers told a different story.

## User Interviews

12 sessions with meeting organizers and team leads across 6 enterprise customers

## Session Analytics

Post-meeting task follow-through rates via App Insights — most tasks were never logged

## Competitive Review

Otter.ai, Fireflies, Notion AI, Copilot for Teams — how competitors handled AI summaries

## Internal Workshops

Design + PM + Engineering alignment on AI pipeline constraints and trust model

### What surprised us

#### Forgetting starts immediately

People couldn't recall key decisions 30 minutes after a meeting. The problem wasn't laziness; it was cognitive capacity under pressure.

#### Manual logging was ritualistic

Those who did log tasks had built their own workarounds: screenshots, emails to self. The tool wasn't supporting their actual habit.

#### Trust in AI was conditional

Users wanted AI-generated output, but only if they could review and edit before it was shared. "Don't make me look bad" came up repeatedly.

# I Reframed the Problem from Logging to Attention

This reframing changed the solution from 'make logging easier' to 'capture the meeting automatically.'

## Before Research

- "Users need a faster way to log tasks"
- "The form is too long"
- "We need better SharePoint integration"
- "Feature adoption is low because onboarding is weak"

## After Research

- "Users can't recall what to log 20 min later"
- "Context disappears the moment a meeting ends"
- "Manual work after a meeting competes with the next task"
- "AI must do the capture, not just simplify the form"

# Before AI, the workflow depended on memory, discipline, and extra effort after the meeting.

## What I mapped from the existing workflow

1

### During the meeting

Organizer manually takes notes or relies on participants to remember what was said

2

### After the meeting

Each task logged individually and synced to Microsoft Planner; name, description, assignee, due date

3

### Decisions recorded

Manually written into SharePoint or sent via email. Often delayed by days or skipped entirely

4

### Summary shared

Written from memory and sent via email. No structured format, no link back to the meeting

## The key failure points

### Tasks fell through the cracks

No automatic capture meant relying on human memory under time pressure. Most were forgotten within an hour.

### Decisions lived in email

No structured record meant decisions were disputed or forgotten within a week. No link back to the meeting.

### No connection between meeting and output

Tasks in Planner had no link back to the agenda item that generated them.

### Organizer burden was unsustainable

The same person who ran the meeting was also expected to document everything — immediately after.

# I Had to Decide What the AI Should Use as Input

## The core design challenge

AI needs input to produce output. But what input? Every meeting is different. My first design challenge was defining which meeting sources the AI should use, and how much control users should have.

### Agenda

Topics, owners, duration

### Tasks

Pre-logged action items

### Decisions

In-meeting logged decisions

### Images

Whiteboard photos, screenshots

### My Notes

Private notes taken  
Pre/During/Post meeting

### Transcript

Auto-generated via Teams and  
Mobile app

## What testing taught us

Early prototypes let users select any combination of sources.

**Testing revealed a problem:** users wanted full control but felt overwhelmed by the options. Choice paralysis was worse than the original manual process.

## What changed:

- Smart defaults - pre-select what's available
- Transcript + Agenda checked when present
- Tasks always included if any exist
- Summary level: Standard vs. Custom
- Meeting type: Virtual / Hybrid / In-Person

I moved from full source control to smart defaults because users needed confidence, not configuration.

# The Final Experience: Reviewable AI Output, Not Automatic Publishing

I separated Recap and Minutes because users had two different needs: a quick shared summary and a more complete official record. Later, the same flow was extended to support in-person and mobile capture as the product evolved.

## Step 1: Select Sources

Smart defaults pre-check available inputs. Users can override, add images, paste notes, and choose Standard or Custom summary level.

## Step 2: Review & Finalize

AI generates structured output: key takeaways, decisions, action items. Users edit inline or apply AI suggestions for tone and detail.

## Step 3: Save & Share

One-click save to the meeting record. Share to all attendees. Recap lives permanently on the meeting — no email required.

## AI Recap

- Compact summary paragraph
- Key takeaways
- Decisions logged
- Action items with assignees
- Shareable to all attendees

## Meeting Minutes

- Full per-topic discussion
- In-depth notes and context
- Complete participant record
- Merge with existing minutes
- Audit-trail quality

## Role Actions

- Organizer: generate, review, publish
- Participant: receive recap + tasks

# Same Meeting. Completely Different Outcomes.

|           | Before AI                                                                                        | After AI                                                                                                 |
|-----------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Tasks     | Manually entered and synced into Microsoft Planner. Fields filled from memory. Often incomplete. | AI extracts action items from transcript, assigns to named participants, syncs to Planner automatically. |
| Decisions | Written to SharePoint List or emailed. No link to the meeting. Frequently skipped.               | Extracted and logged to the meeting record with source context. Permanent, searchable, linked.           |
| Summary   | Written from memory, sent as plain email. No format. Delayed 24-48 hours.                        | AI-generated recap ready within seconds of meeting end. Structured, reviewable, shareable in one click.  |

# The Impact Validated the Design Direction

**+40%**

faster post-meeting  
workflow for organizers

**3×**

more tasks logged per  
meeting vs. manual

**~0**

missed decisions  
in AI-recap meetings

**+28%**

feature adoption  
within first 30 days

## Trust built through control

The Review step was the feature that drove adoption. Users shared publicly only after approving. Removing it in a test caused a significant drop in shares.

## Guests finally included

Share-to-attendees was the most-requested feature post-launch. Meeting outcomes had previously lived inside the product for internal users only.

## AI framing mattered

Naming it 'Decisions AI' rather than 'Auto-Summary' set the right expectation. Users understood it was AI-assisted, not AI-dictated.

*Based on post-launch usage data and workflow comparison.*

# What I Learned Designing AI for Real Workflows

## What surprised me most

My biggest learning was that trust mattered as much as AI quality. Users did not just need accurate output. They needed control before sharing it.

## What I'd do differently

I'd push for user research earlier in the AI prompt engineering phase. We spent weeks tuning output before testing with real users and the first session told us things no internal eval would have caught.

## What the constraints taught me

Being inside the Microsoft ecosystem shaped every decision. Designing within Teams, Planner, and SharePoint forced us to work with integrations rather than around them and that made the product more adoptable, not less.

## What I'd explore next

Guest attendees still don't receive the recap automatically. If a guest has tasks or decisions logged against them, there's no way to surface just that context without exposing the full meeting record. A personalized, scoped recap for guests is the most immediate gap worth solving.

LET'S CONNECT

# Thank You

I'd love to talk about how I can help your team design better products.

[hello@sanjayshrestha.com](mailto:hello@sanjayshrestha.com) · [linkedin.com/in/shresthasans](https://linkedin.com/in/shresthasans) · [sanjayshrestha.com](https://sanjayshrestha.com)

Kathmandu, Nepal · Open to remote or global relocation