KEYNOTE · UXDX 2026

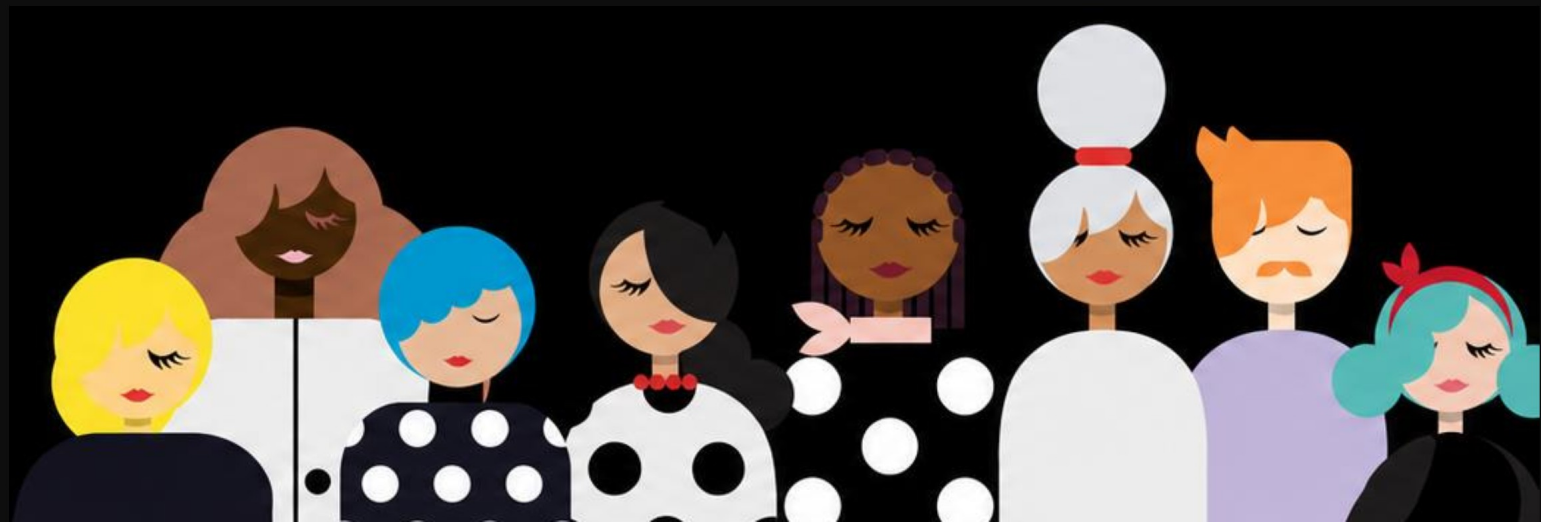
The Intent Steward Model

How Sephora is redesigning product delivery for AI-native teams



SNEHA NARAHALLI

SVP, Product & UX · Sephora



WHERE WE LEFT OFF

Customers arrive wherever their intent begins.

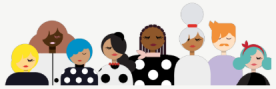
- Search
- Social
- Creators
- ChatGPT & AI assistants
- App
- Web
- Store
- Email

Last time: the customer side

Journeys gave way to intent systems that understand, remember and act in real time.

Today: the inside

What we changed in how Sephora works to build, run and govern those systems.



WHAT WE LEARNT

Eight lessons that reshaped how we work

Context resets at handoffs

Each surface kept its own memory of the client.

Decisions sat in silos

Search, chat and checkout each chose the next step alone.

Coding was never the bottleneck

Faster code exposed slow decisions, reviews and handoffs.

Big documents slow alignment

Requirements locked early were hard to review and slow to change.

Agents need context

They are only as good as the standards and domain knowledge they can reach.

Speed raises risk

Testing, review and security have to scale with code generation.

Small teams need clear owners

Fewer people only works when accountability is explicit.

AI needs a governance home

Pilots moved fast; someone has to set the limits.



WHAT HAD TO CHANGE

Five shifts inside Sephora

Shift	From	To
Ownership	Hand-offs between functions	A steward pair owns each feature
Planning	Feature roadmaps	Features tied to a KPI and ROI
Building	People write every artifact	Agents draft; people decide
Alignment	Big gate meetings	Frequent, smaller check-ins
Measurement	Status reports	Cycle time and post-launch impact



PART ONE

Who Owns the Outcome

Two stewards, one feature, from intent to production.



AGENTIC MICRO PODS

Not every project fits a micro pod

	A micro pod is	A micro pod is not
Scope	One small, well-defined feature, delivered end to end	A large initiative with many dependencies
Team	3–6 dedicated people, QA in from day one, test-first	A long-running pod or roadmap vehicle
Work	AI drafts, humans validate; focused sessions, demos, UAT	A status meeting or presentation forum
Outcome	Usable artifacts, not "completion"	A way to skip review for speed

QUICK TEST If two pods need each other's output to deliver client value, you sliced too thin.

RULE OF THUMB Pods compress time to build-ready work, not the whole SDLC.



THE INTENT STEWARD MODEL

Two steward roles.
One accountable outcome.

THE WHAT AND WHY

Intent Steward

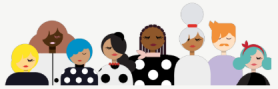
Owns business intent, product definition and acceptance. The business's day-to-day counterpart.

THE HOW

System Steward

Owns architecture, implementation, quality and production health. Directs and reviews agent-written code.

A small pod owns the outcome. Reporting lines stay the same.



HOW A STEWARD WORKS

Every steward runs the same loop.

01

Understand intent

What is the client trying to do?

Query, page, conversation, entry source

02

Apply context

What do we already know?

Beauty Insider, preferences, basket, store, inventory

03

Decide next best action

What should happen now?

Policy, model and guardrails pick one action

04

Measure the outcome

Did it work for the client and the business?

Result feeds the next decision



THE STEWARD'S SOURCE OF TRUTH

The feature brief

Problem

Grounded in client research, scoped to one deliverable feature.

Use cases

Each with an ID, traced through tests and code.

Requirements

Clear enough for agents and engineers to build.

Metrics

The outcome the feature must move, with a baseline.

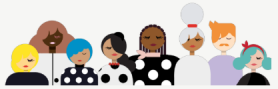
Rollout

Phasing, A/B test plan and launch criteria.

Guardrails

What the system must never do, and when to stop.

Lean by design. Good enough fidelity to align partners, then the pipeline takes it from there.



PART TWO

Operating Model and Guardrails



How we build, run and govern AI that makes decisions for clients.

Seven phases. A human call at each.

ANNUAL PLANNING	AI-SDLC AGENT PIPELINE · STAGES 1–12					LEARN
Opportunity Why should we invest?	**Decompose** Is ROI tied to a KPI?	**Product Design** Real client need?	**Build** Does it match intent?	**Test / UAT** Ready for partners?	**Pilot / Launch** Go or No Go?	**Post-Launch** Did it deliver?

Every phase ends with an exit ticket: a real artifact that proves the work happened.



WHERE UX FITS

Three design paths, matched to ambiguity and risk.

AFTER THE OPPORTUNITY

Foundational research and design

For novel, high-uncertainty work. Builds a shared vision before decomposition.

ALONGSIDE THE BRIEF

Standard design and usability testing

Design, brief and testing refine each other. All platforms designed.

IN THE BUILD · PILOT

Design in the codebase

For simple, low-risk work on existing design-system patterns.

Figma stays the shared canvas for alignment, connected to our design system.



GUARDRAILS · BUILT IN

Six guardrails, in place before launch.

Human in the loop

Agents draft. A named person approves every stage.

Traceability

Use case IDs link the brief to tests, code and the PR.

Automated quality gates

Security, test coverage, performance and SRE readiness.

Council review

Architecture, CyberSec and Privacy clear the approach.

Authority limits

Each agent has a defined action scope and blast radius.

Rollback first

Phased rollouts and A/B tests, with a fast way back.



PART THREE

Working with the Business

When build takes days, the speed of decisions sets the pace.



SET EXPECTATIONS FIRST

What we build stays the same.
How it gets built changes.

STAYS THE SAME

Prioritization: planning, roadmap and funding, in the same forums

Validation: UAT, pilot and Go / No Go with the same partners

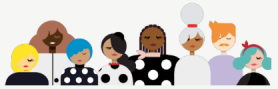
Ceremonies: kickoff, reviews, demos and post-launch reads

CHANGES

Requirements: a lean brief and a working prototype, reviewed often

Decisions: made in the session by an empowered lead

Pace: build to demo in days, timelines in weeks



SAME OPPORTUNITY, DIFFERENT SHAPE

From one big review to many small ones

Before

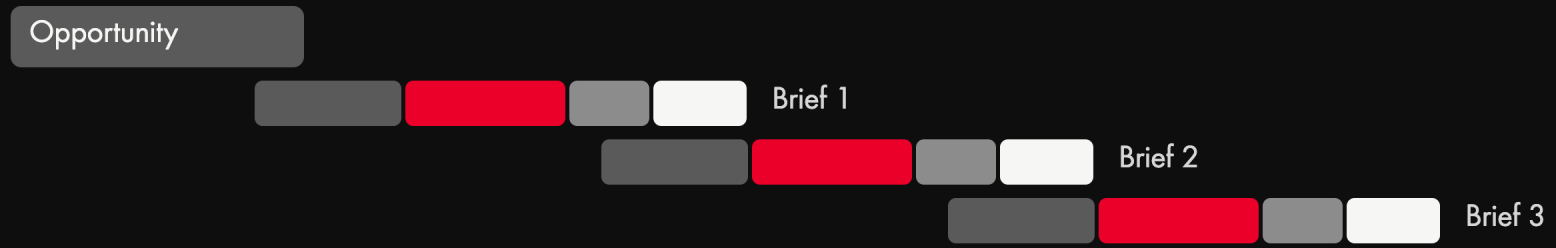
One large PRD, one long build



Business decisions: three, at kickoff, PRD review and discovery exit

Now

Small feature briefs, shipped in sequence



Business decisions: every week, on working software

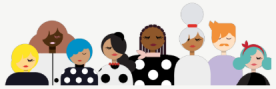
Design Build UAT Launch

TWO KINDS OF MEETINGS

Working sessions do the work. Ceremonies align leaders.

	Working sessions	Product ceremonies
Purpose	Refine the work and decide to move it forward	Business alignment at key milestones
Who	Core team and the named decision-maker	Broad audience, including leadership
Cadence	Set by the work, often weekly	Standing invite on the calendar
Content	Brief refinement, UX reviews, working demos	Kickoff, brief review, design review

The working team signs off first. Leaders then review a decision, not a draft.



THE DECISION LOOP

Decide in the room.
Build the next day.

01

Bring options

The team brings several working prototypes and a pre-read.

02

Pick one

The named business lead decides before the meeting ends.

03

See it working

Agents build overnight. Next session, a working version.

WHAT IT ASKS OF LEADERS

Come ready to decide. Read the pre-read.

Name the decider for each initiative, and be there.

A decision that slips a week slips the build a week.

Decisions can change. Iteration is cheap now.



UAT AND TESTING

Smaller UAT. More experiments. Tighter coordination.

UAT gets focused

One feature at a time instead of a tangle of use cases at the end.

Product tests before business UAT starts.

Scripts and checklists go out as pre-reads.

Testing demand rises

Faster builds mean more changes to validate. Client traffic stays finite.

Ship the smallest meaningful, testable slice.

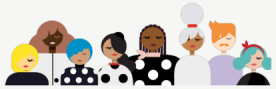
Agree audiences, success measures and guardrails at kickoff.

Ownership holds

The pod owns feature quality, test-first.

QE owns automated, system-wide regression.

Test approval and scheduling stay with the testing team.



PART FOUR · CASE STUDY

One Feature, End to End

Partner Portal: applying the AI SDLC to a real production capability

A small pod moved from intent to working software faster. The bigger lesson was where human judgement still mattered.



THE STARTING POINT

The intent: make partner data easier to see, trust and act on.

THE PROBLEM

Partner-feed status and health were hard for business and SRE teams to see and manage.

USERS

Business users · SRE teams · Partner teams

HYPOTHESIS

One clear place to see feed status will cut friction in onboarding and managing partners, and build a base that scales.

ROUTE AND TEAM

Agentic micro pod · Intent Steward + System Steward · agents doing the specialist work



THROUGH THE AI SDLC

AI did more of the production.
People made the consequential calls.

Phase	Agents accelerated	The human call
Intent	Brief into requirements and criteria	Is this the real problem?
Design	Stories, solution artifacts, options	Does it match the intent?
Build	Code and tests, iterated fast	Do we accept this build?
Quality	Automated tests and quality checks	Is technical quality enough?
Release	Change prepared for deployment	Go or No Go?

The machine produced more. The humans decided more.



THE THREE DECISIONS

The work moved fast because decisions moved fast.

01

What are we actually building?

One place for business and SRE teams to see feed status and content.

Call: set scope and acceptance criteria

02

Is passing the tests enough?

Engineering quality was strong. Human review found a gap the metrics missed.

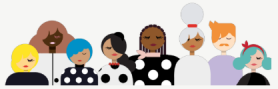
Call: no. Experience fidelity matters too.

03

What changes in the system?

The gap became a lesson for the pipeline, beyond a one-off fix.

Call: change the process, not just the code.



THE MOMENT THAT MATTERED

236 tests passed.
100% code coverage.
And it still wasn't right.

The experience didn't faithfully match the intended design. AI told us the code was healthy. A human looking at the experience told us the product wasn't finished.

WHAT HAPPENED NEXT Design fidelity became an explicit exit criterion.

WHAT CHANGED

The lesson wasn't "AI failed."
The system learned.

BEFORE

Quality = engineering correctness

- ✓ Tests pass
- ✓ Code coverage passes
- ✓ Security passes
- → Ready to release

AFTER

Quality = engineering + experience

- ✓ Tests pass
- ✓ Code coverage passes
- ✓ Security passes
- ✓ Design fidelity passes
- ✓ Human acceptance passes

Every miss should make the next pod smarter.



RESULTS

Faster delivery. High quality.
A better operating model.

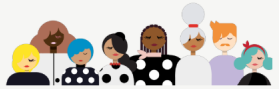
1 release
ahead of the original plan

236
automated tests passing

100%
line coverage

THE RESULT THAT MATTERED MOST

A missing quality criterion, found and permanently added to the system: design fidelity.



WHAT THIS TAUGHT US

Four lessons for AI-native product development

01 Speed moves the constraint

Once production gets faster, decisions become the bottleneck.

02 Tests prove what you specify

They prove technical correctness. Experience quality needs judgement.

03 Humans move up the stack

Less time producing artifacts. More time judging intent, trade-offs and outcomes.

04 Learning changes the system

No model prevents every miss. A good one makes sure you never repeat it.



SO WHAT ACTUALLY CHANGED?

AI didn't remove the team.
It changed the team's job.

Old SDLC	AI-native SDLC
Humans produce artifacts	Agents produce artifacts
Humans coordinate handoffs	Agents carry context
Humans wait for reviews	Automated gates run continuously
Decisions at milestones	Decisions on working software
Success = shipped	Success = outcome + learning

The scarce resource is no longer production capacity. It is judgement.



TAKE HOME

If you remember three things

1

Give every outcome an owner.

Not a committee. Not a function. A named human being.

2

Make context executable.

Requirements, design rules, architecture, tests and guardrails become context agents can use.

3

Let AI earn autonomy.

Automate what can be verified. Keep humans in the loop where judgement, risk and experience matter.



CLOSING

In a world of exploding complexity, our job is to make it feel invisible.

Design around intent. Give every outcome an owner. Let AI act only as far as it has earned.



THANK YOU

Sneha Narahalli · SVP, Product & UX · Sephora