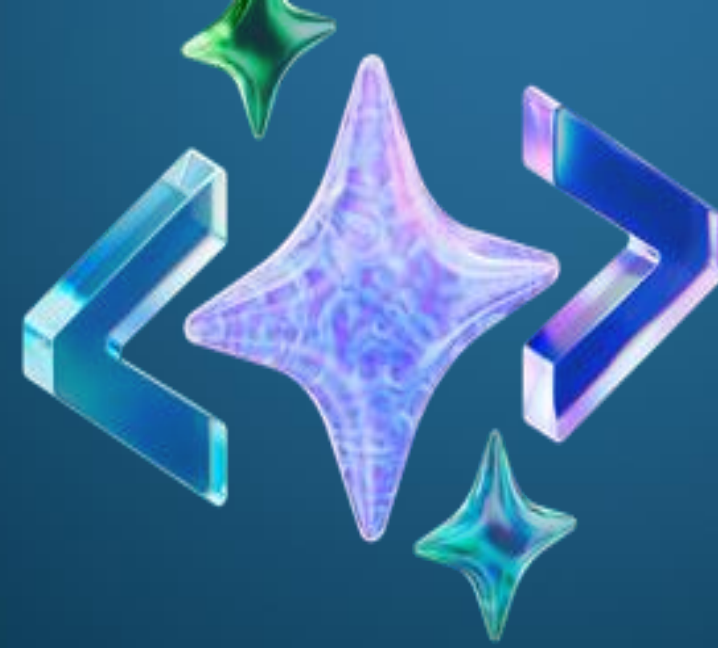


AI is rewriting the Product Lifecycle



Prabodh Ambale

GVP, CRM & Industry Workflows Engineering
ServiceNow

Four Transformational Shifts Underway

AI is changing what we build, how we build it, how we release it, and how we land it



Product Transformation

The application stops being a place people log into and becomes something that works alongside them.

UI led → AI led

- Simple work needs no login—autonomous agents and front-door tools handle it
- The product becomes the system of action, not the destination

Conversation first

- Voice, chat, and messaging as the primary interface for most users
- The full interface is there when the work gets complex

AI in every facet

- Admin setup, design time, and runtime all AI assisted
- Users ask for what they need in their own words

Built to move fast

- An open source heavy microservices stack underneath that AI tools are proficient with
- Restructured product metadata for efficient AI tool generation and updates

Every phase is compressing

AI is reshaping how we plan, build, deliver, and deploy—and harness engineering is doing much of the heavy lifting

Plan
sprints → days

Intent becomes scoped,
estimated work in a sitting

Build
weeks → days

Agents draft the code.
Engineers own the decision
gates

Deliver
days → hours

Tests, reviews, and docs
arrive with the change, not
after it.

Deploy
days → minutes

Ship on demand, with
rollout and rollback
guarded by default.

Harness engineering

Teams of **AI Agents** orchestrating the end-to-end PDLC – ideation, discovery, prototyping, technical design and architecture, critique, code generation, test automation, code reviews and code merge.

Product **domain context** underpins and makes every Agent in the harness more efficient and accurate

Controls and **Governance** part of the SDLC including testing (token spikes are product regressions)

Releases get smaller and faster

As the cadence compresses, how we plan, partner, and upgrade changes with it.

Semi-annual → Quarterly → Monthly → Weekly

Planning decentralizes

Teams own their own release path and move at the pace their work allows.

Dependencies by design

Teams come together for a fixed window to execute shared priorities, cutting straight across org lines.

Customers built in

Design partners and early adopters ship with us; general availability comes once customers are proven successful.

Transparent upgrades

Backward compatible by default, with zero upgrade effort asked of customers.

Engineering moves to the field

Forward deployed engineers make deployment about outcomes and value realized, not features shipped.

Discover → Scope → Configure or build → Test → Deploy

Accelerated by harness engineering—tooling for use case discovery, scoping, and execution, built the way we build products.

From features to outcomes

Deployment succeeds when customers realize value, not when the feature ships.

Embedded with partners

Product and engineering work alongside implementation partners and professional services, from discovery through go-live.

A direct closed loop

What engineering learns flows back into the product.

Integrated Controls & Governance

Deployments are deeply integrated with Governance tools to monitor Agentic identities and authorization, usage, spend, value realized (time savings, cost savings)

Skills and Mindset Transformation

The transformation is real—and so is AI slop

Everyone reviews

Judgment calls on AI output become the job. Hardest for entry-level engineers, even as they arrive more AI native.

Everyone manages

A team of agents to run: knowing what works, and fixing harnesses, agents, and skills when it doesn't.

Bandwidth stretches

Velocity keeps climbing. What a mind can hold at once does not.

Managing outcomes

Engineers now carry deployment outcomes, not just merged code.

Reskilling and upskilling is an immediate imperative

This is the partially resolved part of the transformation, and it will test all of us

Thank you