

From Prototypes to Production Agents



What Actually Matters

Cansu Berkem
Director of Product Management
Datadog

About Me

Cansu Berkem, Director of Product Management @ Datadog

15 years in Product Management

- Salesforce - Marketing Cloud
- Salesforce - Customer Data Platform
- Datadog - Bits AI
- Datadog - Agent Builder & Agent Console



What We'll Cover



Autonomy First

How much should agents act on their own?



Flexible Reasoning

From fixed workflows into adaptive reasoning



Design For Failure

Making uncertainty explicit and honest



Transparency & Trust

Branching hypothesis users can follow



The Feedback Loop

One word that unlocked everything



Production Realities

Context limits, multi-agent

The Foundation

MCP has made it dramatically easier to build AI Agents. Teams go from ideas to working prototypes in days



Clean Interfaces

Well defined
integration between
models and tools



Fast Iteration

Test and iterate
without heavy setup



Defined Boundaries

Clear limits on agent
actions to ensure
safe and predictable
behavior

Autonomy is the first decision

Autonomy is not binary,
shape it carefully based on:

- Risk level of action,
- Clarity of the context,
- User expectations,
- Regulatory requirements

The Autonomy Spectrum

Full Manual

Human does everything

Assisted

Agent suggests, human acts

Supervised

Agent acts, human approves risk

← Bits AI SRE

Full Auto

Agent decides and executes

Reasoning beyond fixed workflows

Before: Fixed Workflows

Gather Context



Run Predefined Queries



Summarize Results

*Broke down on multi-service failures,
missing context*



After: Adaptive Reasoning

Generate Hypothesis



Test and Validate each



Explore new signals
dynamically

*Investigates like a Senior SRE, flexible,
iterative, collaborative*

Design for Failure, Not Just Success

Ambiguity is everywhere in production. Biggest mistake is assuming the agent will always find the right answer.



Conclusive

Agent is confident in its finding. Root cause identified with supporting evidence.

Charged · High value · Builds trust



Inconclusive

Multiple possible causes, missing data, or conflicting signals. Uncertainty is surfaced.






Not charged · Honest signal · Drives improvement

Transparency Builds Trust

Branching Hypothesis System



Why It Matters

-  Users can follow the investigation step by step
-  Evidence is visible — not a black-box answer
-  Multiple paths explored and shown
-  Feels like working alongside another engineer
-  Builds confidence in both results and non-results

The Feedback Loop

The most impactful change had nothing to do with the model

No

Did Bits find the answer?

Nobody clicked.

Not Quite

Changed one word

Feedback flooded in.



Match Engineer Thinking

“Not quite” reflects how real people think about partial answers



Real-Time Pulse

Every response streamed to Slack — a live performance dashboard



Granular Feedback

Embed feedback at each hypothesis, not just at the end

When Do You Actually Need Humans?

High Risk
+ High Clarity



Human approval required

Modifying infra, payments, medical

High Risk
+ Ambiguous



Definitely need humans

Unclear billing issues, security events

Low Risk
+ High Clarity



Automate freely

Sending notifications, tagging tickets

Low Risk
+ Ambiguous



Clarify first, then act

Vague user requests, missing context

Key Takeaways

Shape autonomy carefully, match it to risk, context and expectations

Flexible hypothesis driven reasoning outperforms fixed workflows in messy reality

Surface uncertainty honestly. Conclusive vs Inconclusive builds more trust than false confidence.

Feedback loops are gold, build your evals around real human insights.

Thank You!