QCon AI Boston 2026

AI First, Quality Always

Agentic SDLC Adoption – A Case Study from Red Hat AI

Catherine Weeks
Director, Agents & AI Eng Tools – Red Hat AI

Red Hat

Catherine Weeks
Director, Engineering
Red Hat AI

Red Hat AI Organization

1 of 5 Engineering Leaders supporting a 600+ person organization.

Platform Strategy: "From Metal to Agent"

Building an AI Platform for customers, bringing GPU and inference efficiency through AgentOps to help enterprises deploy and manage AI successfully.

Today's Focus: Internal Transformation

Consuming AI to transform our own engineering organization.

Red Hat

Act 1

The Pressure

The "AI First" Mandate

What it looks like when it lands on a 500+ person engineering org

Executive Urgency

The urgency is real and rational — AI is reshaping competitive dynamics in every industry.

Structural Gaps

Mandates without proper structure create perverse incentives and organizational friction.

Measurement Trap

The pressure to show results fast leads teams to measure the wrong things, prioritizing output over value.

"How many PRs do you want? I can have the genie write the script to chop up a PR to achieve any number that you want."

— Kent Beck

The Brand Risk

What happens when you optimize for speed over quality

Technical Failure

AI-generated code that passes CI but fails in production

Operational Erosion

Escaped defects, security regressions, maintainer fatigue

Asymmetric Cost

Customer trust erodes slowly and recovers slowly — the cost is asymmetric

Brand Damage

The organizations that skip guardrails to hit timelines will pay in brand damage

Roles in AI Evolution

Engineering Leader

What happened:

- Dragging feet on adoption / Ignoring
- Jumping in too fast with one team without building agreements for org

If this is you, what is your role?

- Understanding org direction
- Acknowledging market change
- Exploring possibilities openly
- Hands on experience
- Leading through change

Architect / Staff Engineer

What happened:

- Fighting AI Automation as “not feasible” and “overzealous execs”
- Getting assigned to drive AI SDLC efforts

If this is you, what is your role?

- Driving best practices for the organization
- Helping build AI Automation that helps the organization succeed
- Mentoring engineers through change

Software Engineer

What happened:

- AI adoption curve active: Innovators to Laggards

If this is you, what is your role?

- Providing input, innovation, and feedback
- Understand the speed of change, and that being a laggard is very risky in this environment.

Act 2

Our Story, and a Framework

Two Modes of AI Adoption

AI CO-PILOT

Engineers use agents to augment existing workflows

Engineers review code, make judgment calls, submit PRs

The goal: Mature engineer's usage of AI.

A good reference is Yegge's 8 Levels of AI adoption from no AI through building your own orchestrator to manage many agents.

AI FACTORY

Fully automated agentic software development

Agents produce implementations from specs and validate results

The goal: can agents perform the full SDLC for simple tasks, pushing the human element to work on more challenging problems?

The models will never be **worse** than they are today.

The target for what can be automated is always **moving forward**.

Three Layers of Adoption

Each layer has different problems and measures of success

1. ORGANIZATION

Standardize your SDLC, then automate it

- Staff a dedicated team per capability area
- Phase by evidence of success, not executive timeline

2. TEAM

Context-specific agentic workflows

- Org standards set the floor; teams own the nuance
- Variability in AI solutions is a feature, not a bug

3. INDIVIDUAL

Personal productivity that compounds

- The most visible layer, but the least structured
- Must connect to team and org quality standards

Layer 1: Organization

Standardize, Then Automate

Every org has SDLC standards. Most are inconsistently applied or ignored.

Decompose SDLC

Identify discrete capability areas:

- Requirements & Architecture
- Implementation & Testing
- Security & Documentation
- Build & Release

Risk Assessment

Analyze human performance & impact:

- Identify human underperformance
- Define "blast radius" for each area
- Differentiate recoverable vs. catastrophic AI errors

Tiger Team Model

Dedicated focus per area:

- One specialized team per capability
- Independent maturity timelines
- Evidence-based progression

THE PAYOFF: Standardized organizational processes automated & clear for agents and humans

Our Example: Requirements Pipeline

Fully autonomous agent managing the RFE lifecycle

Review → Split → Fix → Submit — human-in-the-loop only for unrecoverable errors

The numbers (week 7):

- **143** features processed through automated pipeline
- **97** RFEs cleared automated review (**54** auto-created, **45** auto-revised)
- **12** oversized RFEs auto-split into **44** well-scoped children
- **~30%** flagged for human attention — that's a feature, not a failure

Evolution:

MVP (week 1) → Autonomous CI (week 2) → PM testing (week 4) → Org-wide enablement

Org Example: Strategy + Security Pipelines

Strategy Pipeline

- **Implementation Strategies:** Generated from approved RFEs
- **Multi-dimensional Review:** Scope, feasibility, testability, security, architecture
- **Live Automation:** 143 features processed via Jira
- **Human Review:** Line-by-line feedback via Git PRs

Security Review Pipeline

- **Optimization:** 58% false positive rate reduced to 22%
- **Scale:** 4 iterations across 1,000+ real issues
- **Impact:** Identified and fixed 6 structural problems
- **Scope:** 8 security dimensions evaluated
- **Quality:** Zero duplicate findings

Meta-lesson:
Agent quality requires the same engineering discipline as product quality

Security Review Pipeline Optimization

False Positive Rate Reduction

58% → 22%

Scale

4 iterations across 1,000+ real issues

Structural Impact

6 structural problems identified and fixed

Layer 2: Team

Context-Specific Workflows

One-size-fits-all agentic adoption breaks on contact with reality.

Codify Context

Embed team-specific knowledge into agentic context:

- Repos & tech stacks
- Ceremonies & rituals
- Tribal knowledge

Skills Marketplace

Enable organic adoption across the team:

- Teams publish skills
- Engineers install/pull
- Pull beats top-down push

Quality via Variability

Different perspectives catch more than one standard:

- Diverse reviewer perspectives
- Avoid "monoculture" blind spots
- Standardization vs. Nuance

THE PAYOFF: Team-level adoption is where AI earns developer trust

Team Example: Skills Registry & Adoption

Skills Registry: The Bridge

Curated marketplace for org standards and team workflows. Features CI validation, schema checks, and security scanning.

QE Tiger Team

- Published 15+ skills (PR/Risk/Quality assessments)
- Viral organic adoption before any mandate

Architecture Context

- Solved hallucinations by moving to YAML
- "The team closest to the problem solved it"

Eval Harness: Cross-Team Success

Built by RFE team, now widely adopted by QE and other teams for their own custom skills.

Layer 3: Individual

Productivity That Compounds

Individual engineers integrating AI into daily work

Daily Integration

Focus areas for AI support:

- Code generation
- Investigation
- Documentation
- Context management

The Risk

Avoiding isolation:

Individual productivity that fails to connect to team or organizational standards.

The Opportunity

The compounding effect:

When all three layers align, the efficiency gains are enormous across the org.

THE PAYOFF: AI Enabled engineers that grow and evolve with the market change

Individual Example: Yolo → Orchestrator

Yegge's Stage	What It Looks Like	The Engineer's Move
3	**YOLO mode – trusting the agent to run**	Starting point (already past "ask it questions")
6	**3–5 agents in parallel, multiplexing**	Built containerized sandboxes to manage a fleet of agents securely
8	**Building your own orchestrator**	Fire-and-forget workflow: assign task → agent works autonomously → harvest commits → open PR

> Used AI to build the tool that manages AI – then shared it across the org so others could climb the same ladder.

The tool started as personal productivity. He presented it to multiple internal groups, recorded demos, and helped others understand how to move up the maturity ladder themselves.

Act 3

Guardrails as a competitive advantage

Guardrails Are Features, Not Friction

Automated Review

Multi-dimensional automated review running in parallel, not serial.

Scope · Feasibility · Testability · Architecture · Security

Eval Harness

3-layer scoring with regression detection. Test your agents like you test your product.

Discovery: Model A self-corrects bugs while Model B surfaces them.

Human-in-the-Loop

Gates at every stage. Agents that know when to pause > agents that never stop.

RFE review → Strategy sign-off → PR review → Merge

Parallel Security

Security is a parallel concern, not a deferred one.

- Skills registry security scanning in CI
- Treating skill supply chain like software supply chain

"To go faster, you need higher quality." – Martin Fowler

> *"To go faster, you need higher quality. Quality drives productivity in the software industry. I don't think agents are going to change this reality."*
>
> — Martin Fowler

Act 4

What you can do tomorrow

What You Can Do Tomorrow

ORG

1. Pick your biggest SDLC Headache

One whiteboard session with your tech leads to automate with AI

ORG

2. Have the metrics conversation now

Explain why speed metrics are dangerous. Don't wait until the wrong scorecard is in place.

TEAM

3. Gathering context is king

The team that volunteers first will teach you the most into agentic context.

INDIVIDUAL

4. Lower the barrier to experiment

But connect tooling to team and org quality standards.

ALL LAYERS

5. Build the escalation path & evaluation harness before you build the automation

Agents that know when to stop > agents that never do.

AI First, Quality Always

Strategic principles for agentic automation and responsible AI scaling.

Key Takeaways & Learnings

ADOPT

1. Adopt across three layers

Org, team, and individual layers each require a unique playbook as they mature at different rates.

MEASURE

2. Measure quality, not output

Expect your first metric design to be wrong; build feedback loops early to refine your approach.

TRUST

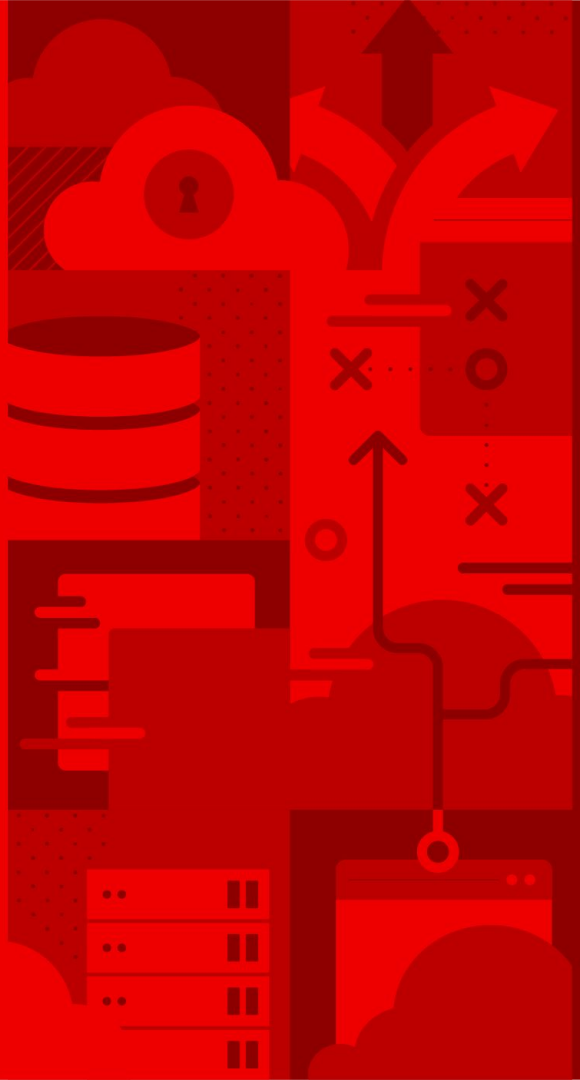
3. Guardrails are features

Responsible adoption provides a competitive advantage; customers prioritize trust over raw speed.

FUTURE FOCUS

4. AI will only get better

Continuously push the boundary of automation; use co-pilots for value today while you build for factories next.



Thank You

Catherine Weeks

Director, Agents & AI Eng Tools – Red Hat AI

@catherineweeks

linkedin.com/in/catherineweeks

Red Hat