Enterprise AI Summit · San Jose · April 2026

Lessons from Transforming Teams to AI-Native Workflows

Tim Cochran

About Me

- Grew up at Thoughtworks
- Joined Amazon’s Software Builder Experience (ASBX) two years ago
- Initially improving AI Adoption, then piloting AI-Native workflows with organizations that are embracing AI
- Co-founding a startup to drive AI effectiveness for enterprises

What does AI-Native mean?

Reinventing the entire product development workflow with AI at the center.

In practice:

- AI agents do the heavy lifting on low-value coding, maintenance, and operations tasks, while the developer directs and supervises.
- AI Agents assist technologists to ideate, design and build.
- *What else?*

Lesson 1: Empower teams: Through trusted leaders, building blocks & community

1) Empowering teams

Managers and trusted senior technical leaders are key to enable teams.

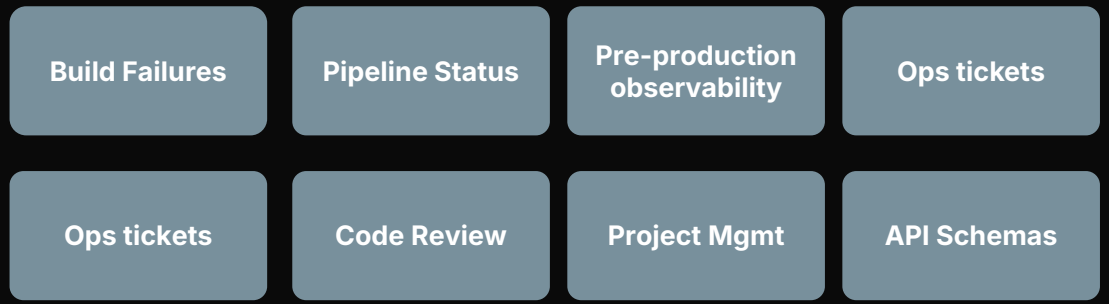
They are often neglected with training. We educated on:

- Everyone has misunderstanding and misgivings, how to help their team
- Fundamentals & the art of the possible.
- DevEx & productivity metrics: *suddenly **a lot of** people are interested in this*
- Creating an AI-First Culture
- Expectations of impact

1) Empowering teams

Create the building blocks for developers to experiment.

- Seamless agent access to develop, build, release and operational tooling:
- Fast discovery and retrieval of knowledge, optimized for Agents.
- Understand and optimize (with AI) the outcomes of the small JTBD the developers do frequently.



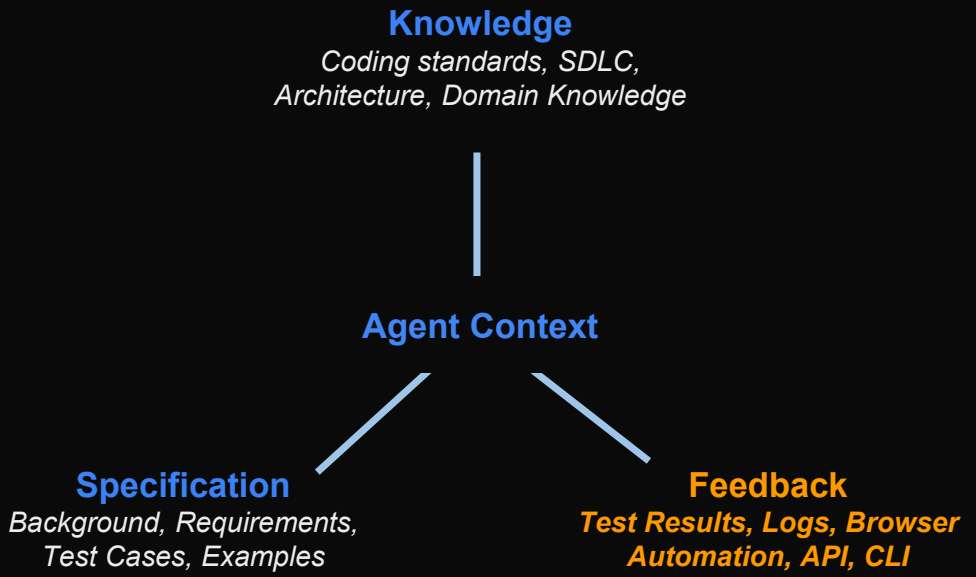
Lesson 2: AI is massive paradigm shift: developers will take time to master AI skills through continual practise

2) Mastering AI Skills

Learn in the context of the team

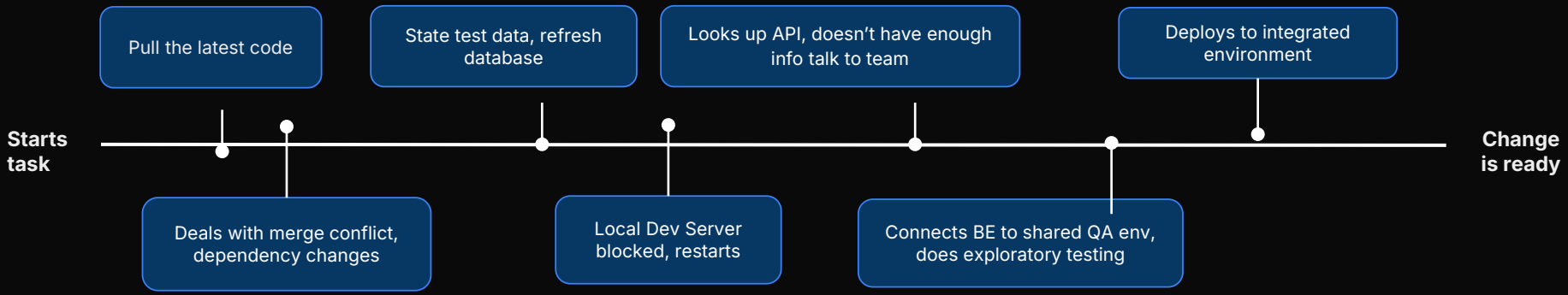
- ***The journey*** is more interesting than demos: observing teammates failing and trying again.
- Create opportunities for developers to observe: Pair Programming and Mob Programming
- Leverage early adopters as champions, foster organic power user community

Understanding context is a fundamental skill



Antipattern: Hand feeding the Agent

Without a good harness. The developer is **hand feeding the agent**, providing perfect little morsels to the agent to consume. Instead we have to teach **the agent how to feed itself**



Lesson 3: Coding isn’t the only bottleneck: apply AI to friction across the product value stream

Two Customers

Your brownfield developer experience was built for humans, not agents.

Existing developer environments are **full of papercuts & bottlenecks** that developer's perform workarounds for.

Every gap a developer learned to live with becomes a **hurdle the agent must overcome.**

3) Product Value Stream

AI Readiness Litmus Check

Take something from your backlog, ideally small and deterministic. Observe a developer and note what it takes to make the change production ready.

Tells us limitation in the teams setup and platform tools.

Record:

- **Steps and Time**
- **Developer Interventions** – everytime a developer has to do something
- **Agentic Friction** – when an agents complete an action, but not in the most efficient way

3) Product Value Stream

AI Readiness Litmus Check - Extension

- Debug a problem
- Triaging an oncall page
- Onboarding: an agent working on a legacy codebase, that has not been worked on before by agents
- Doing a more full fledged feature, multi-PR, track from ideation

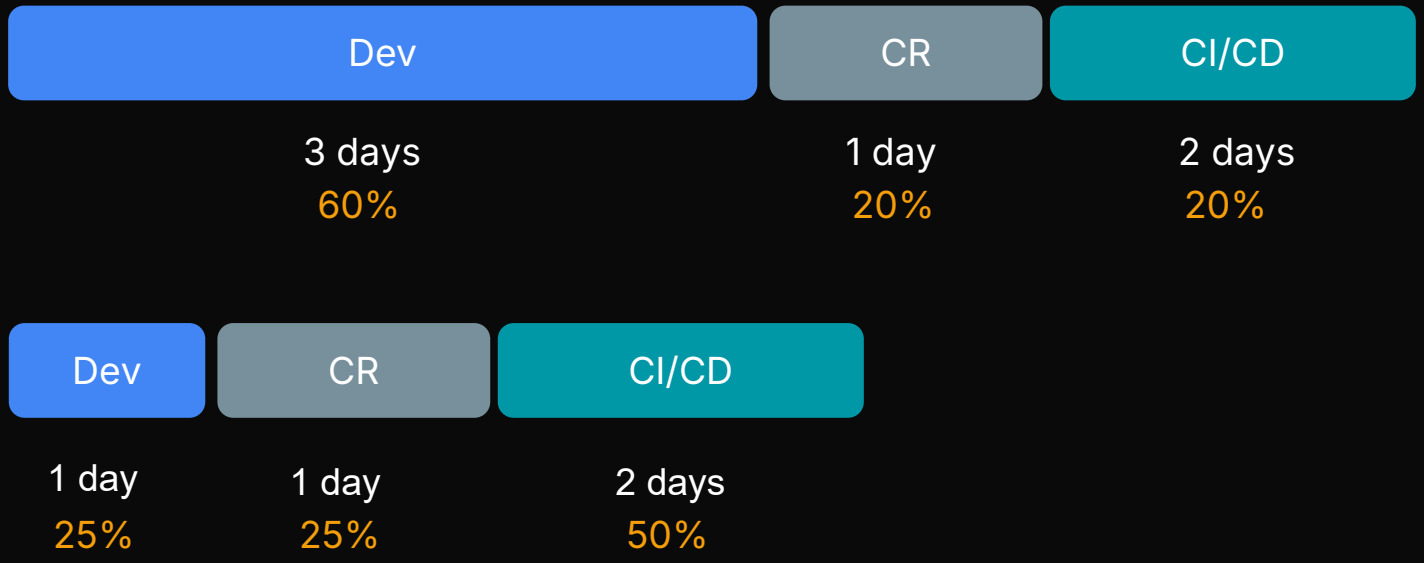
3) Product Value Stream

Applying AI to common friction

DESIGN	BUILD	RELEASE	OPERATE
Paring with the Agent the flesh out a idea	Agents trained on enterprise knowledge	Project Status reports semi-automated	Generate Oncall reports
Writing documents in the required format for Org	Maintenance tasks e.g. upgrade dependency	Common builds failures are auto fixed	First Pass Root Causes Analysis for issues

Lesson 4: AI Amplifies your existing bottlenecks:
Apply High-throughput engineering best practises

Friction previously deprioritized is now a bottleneck

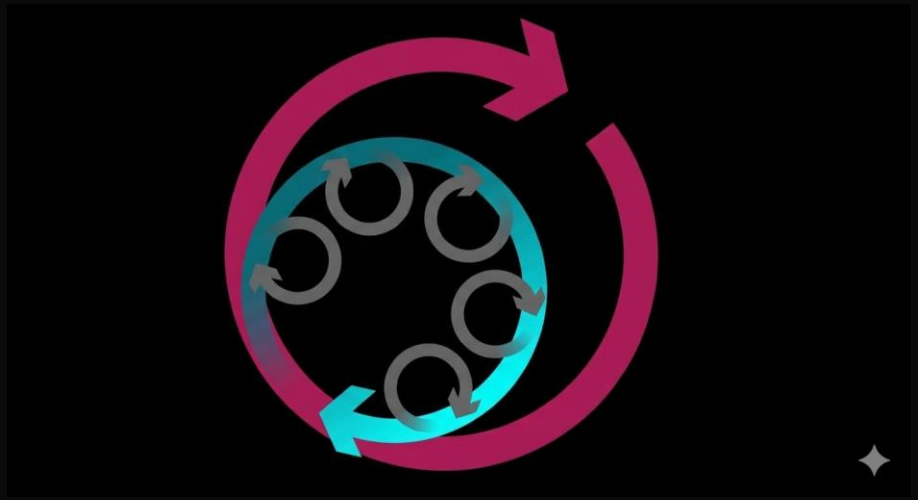


Patterns of High Throughput teams

- Continuous Deployment
- Small Atomic Changes
- Optimize onboarding and knowledge discovery for Agents
- Modularized codebases
- Agents can solve their own problems – debug, fix builds, exploratory tests.

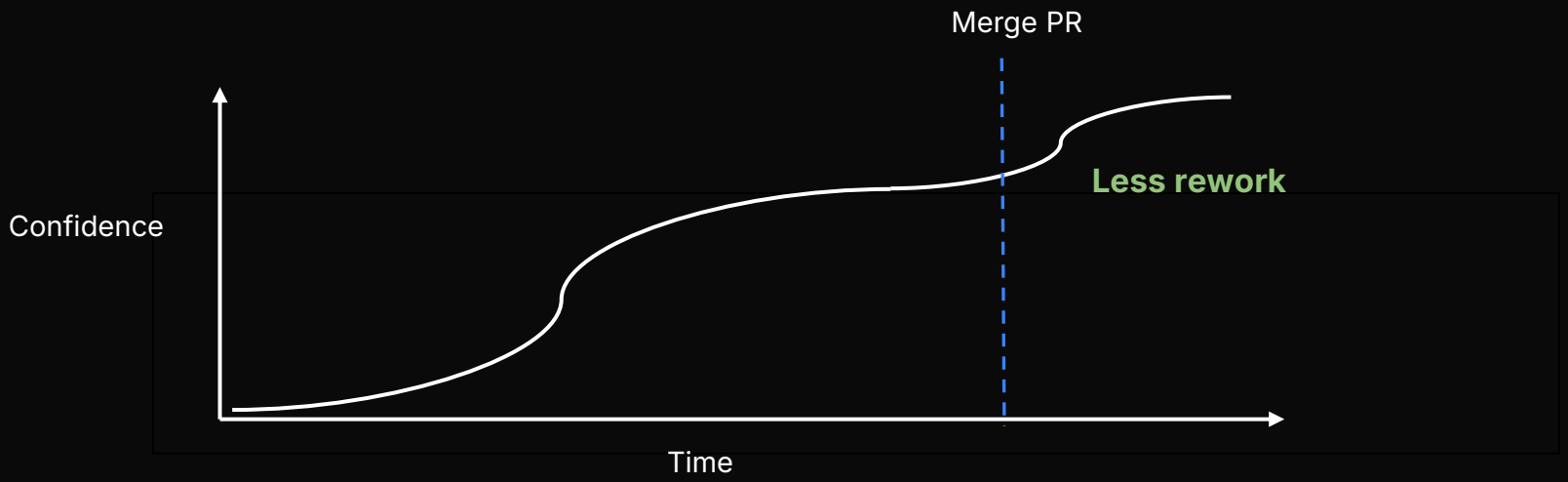
4) High-throughput team patterns

The most effective teams can quickly check changes in a prod-like environment.



https://martinfowler.com/articles/developer-effectiveness.html

High confidence before merging



After 20 years, agents might finally force us to shift left!

Lessons

1) **Empower teams:** through trusted leaders, building blocks & community
2) **AI is massive paradigm shift:** developers will take time to master AI skills through continual practice
3) **Coding isn’t the only bottleneck:** apply AI to friction across the product value stream
4) **AI Amplifies your existing bottlenecks:** Apply high-throughput engineering best practises