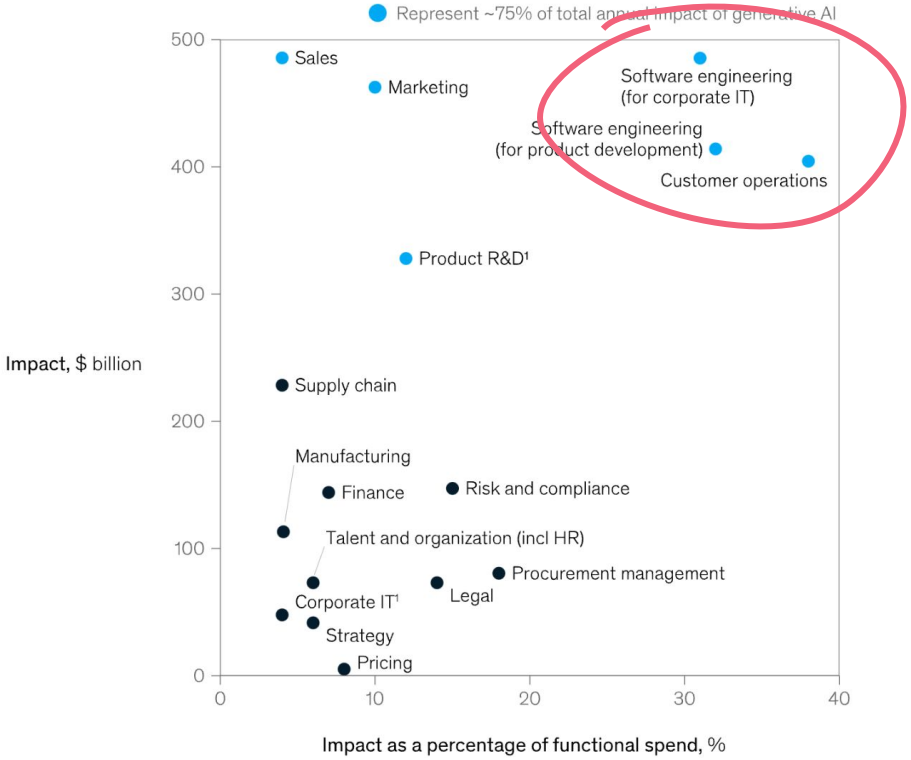
The power of AI in software delivery



/thoughtworks

McKinsey predicts software engineering will be the business function most impacted by Generative AI

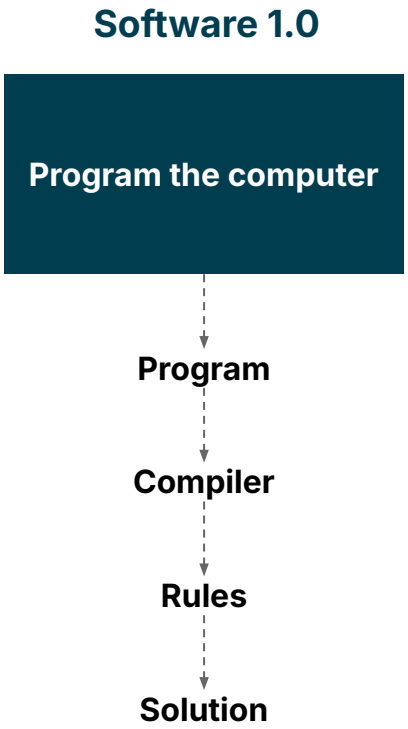


Note: Impact is averaged.
¹Excluding software engineering.
Source: Comparative Industry Service (CIS), IHS Markit; Oxford Economics; McKinsey Corporate and Business Functions database; McKinsey Manufacturing and Supply Chain 360; McKinsey Sales Navigator; Ignite, a McKinsey database; McKinsey analysis

McKinsey & Company

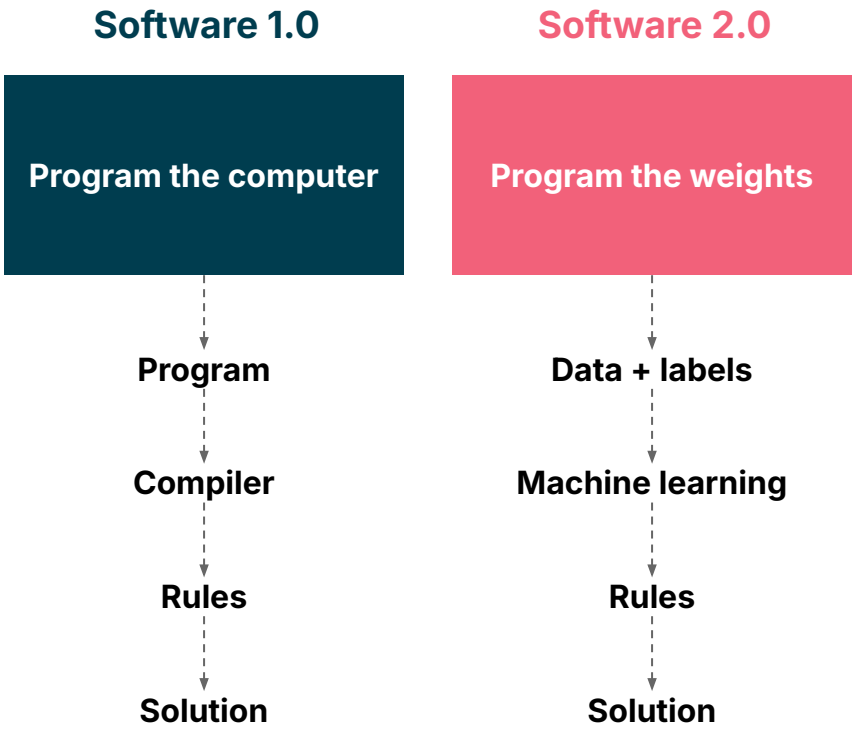
© 2025 Thoughtworks | Confidential

The evolving software engineering paradigms



Credit: Andrej Karpathy

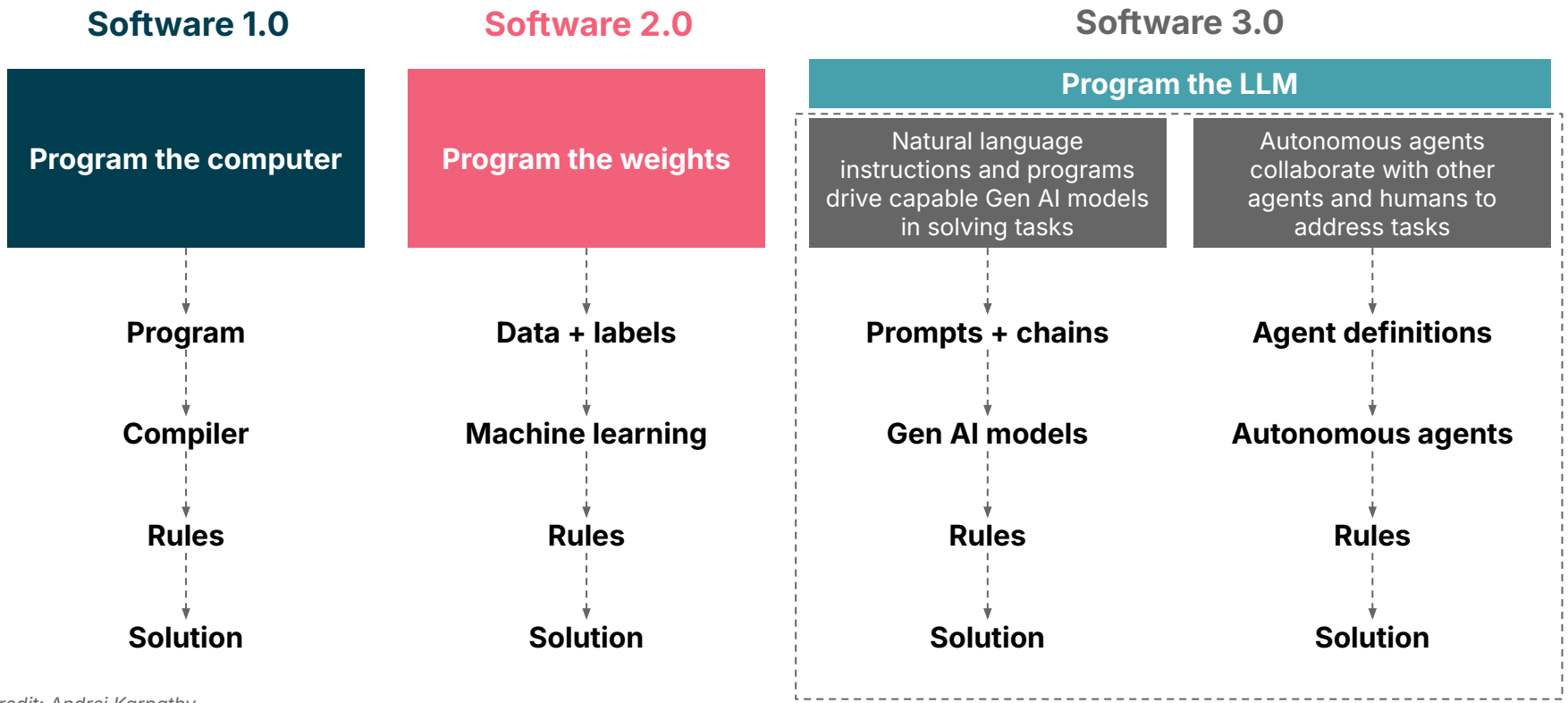
© 2025 Thoughtworks | Confidential



Credit: Andrej Karpathy

© 2025 Thoughtworks | Confidential

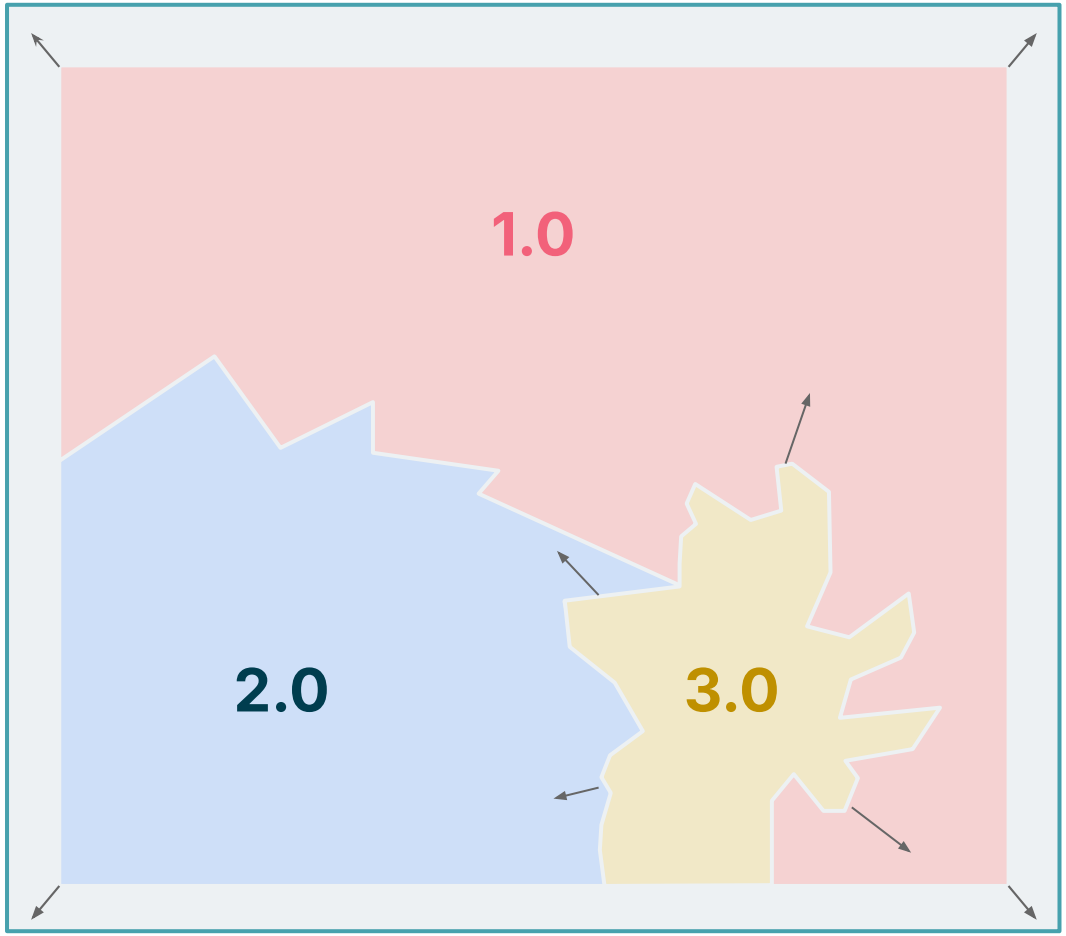
AI is changing the way we write software



Credit: Andrej Karpathy

© 2025 Thoughtworks | Confidential

More and more software will be written in 3.0



Credit: Andrej Karpathy

© 2025 Thoughtworks | Confidential

AI adoption & scaling challenges



/thoughtworks

© 2025 Thoughtworks | Confidential

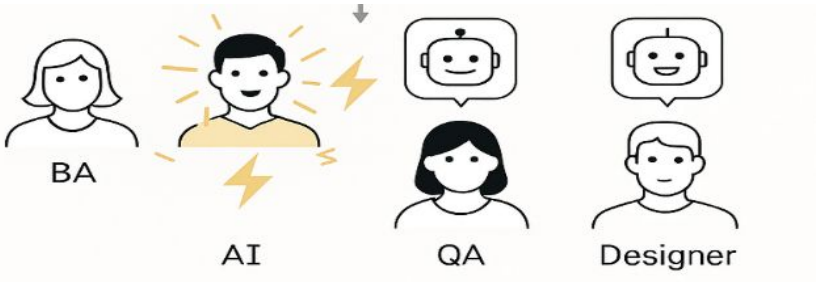
#1. Tool fragmentation & workflow disruption

/thoughtworks

© 2025 Thoughtworks, Inc. Commercial in Confidence

Problem 1: Imbalanced AI adoption

Only certain roles (e.g. Developers) are enhanced by AI, while others remain under supported



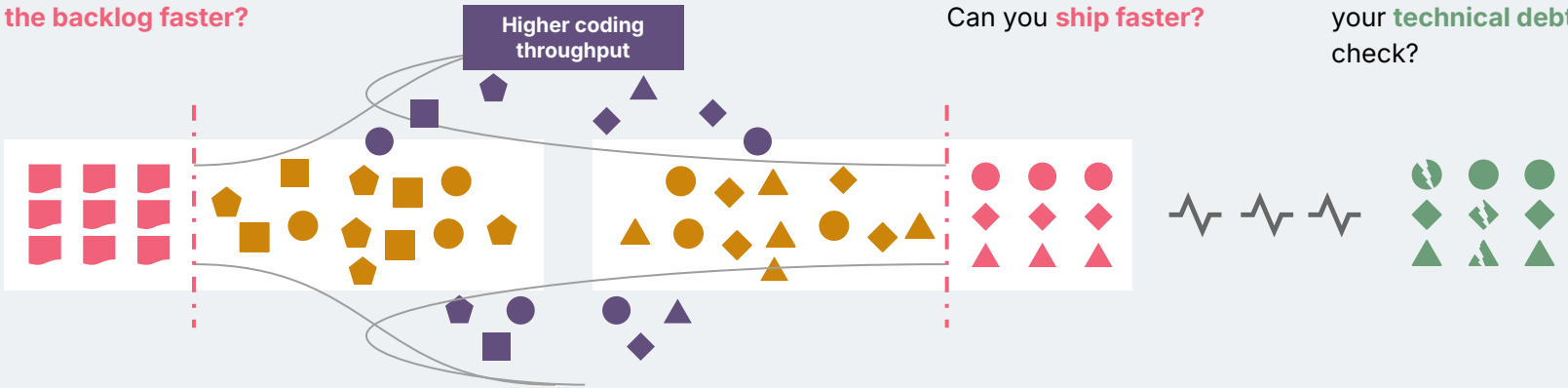
© 2025 Thoughtworks | Confidential

Higher coding throughput starts putting pressure on the system.

If you can code faster, can you **fill the backlog faster?**

If you can code faster, Can you **review faster**? Can you **ship faster?**

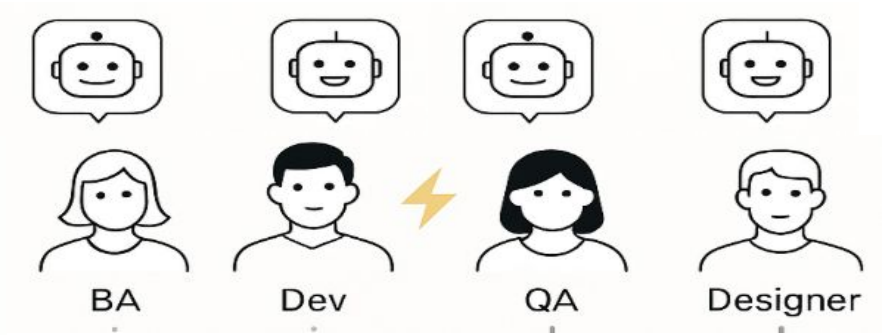
If you can produce more code, can you also keep your **technical debt** in check?



© 2025 Thoughtworks | Confidential

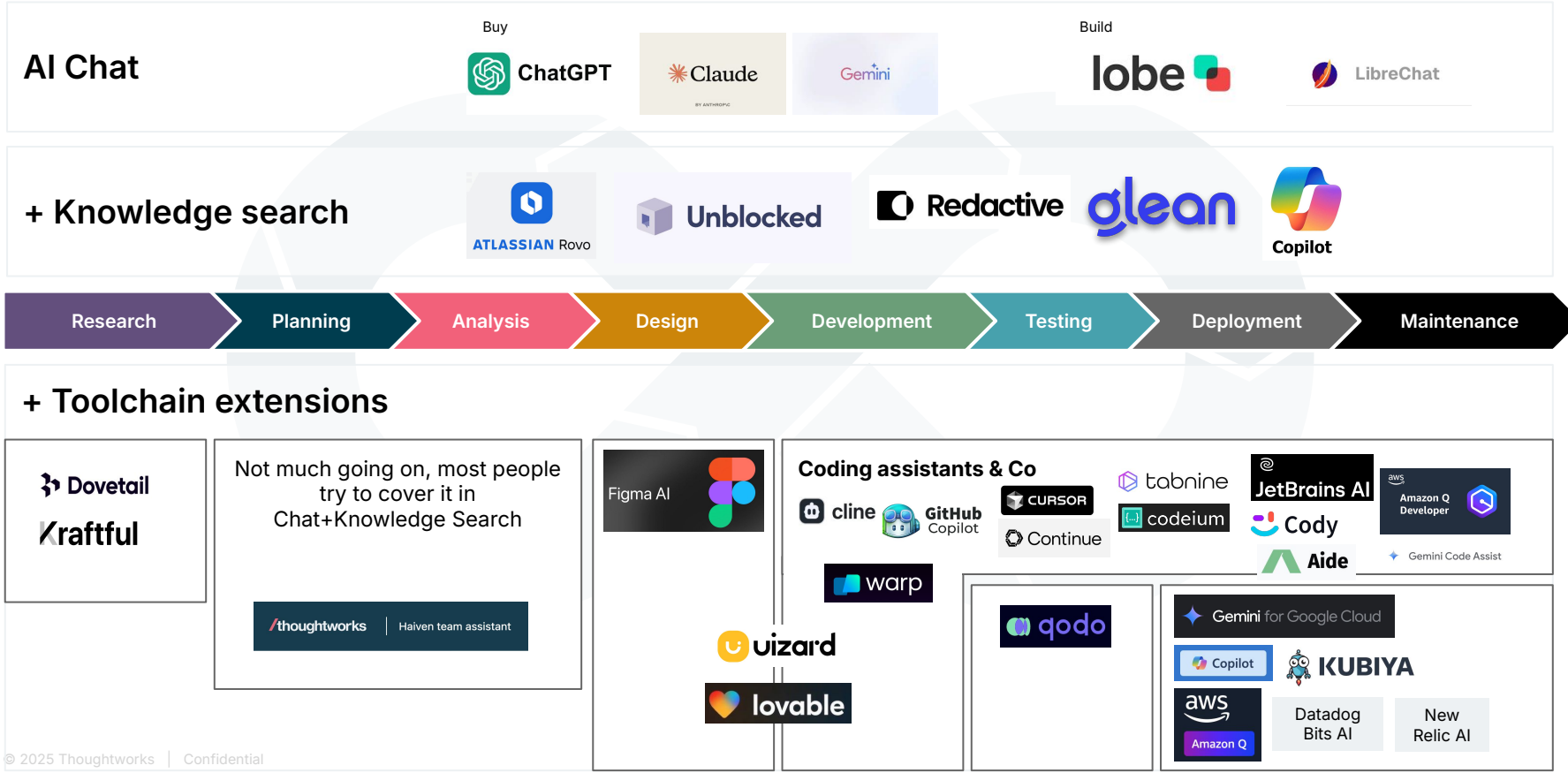
Problem 2: AI enhances roles but not their relationships.

AI assistants operate in isolation; they don’t collaborate across roles



© 2025 Thoughtworks | Confidential

The overcrowded tooling ecosystem

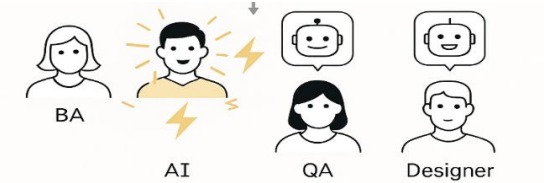


© 2025 Thoughtworks | Confidential

AI is supercharging the role; not the TEAM

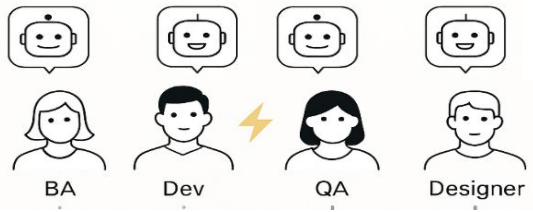
Problem 1: Imbalanced AI adoption

Only certain roles (e.g. Developers) are enhanced by AI, while others remain under supported



Problem 2: AI enhances roles but not their relationships.

AI assistants operate in isolation; they don't collaborate across roles



Result: Local speed increases, but system-wide delivery still drags.True acceleration requires AI to improve not just individual tasks — but how roles work **together**.

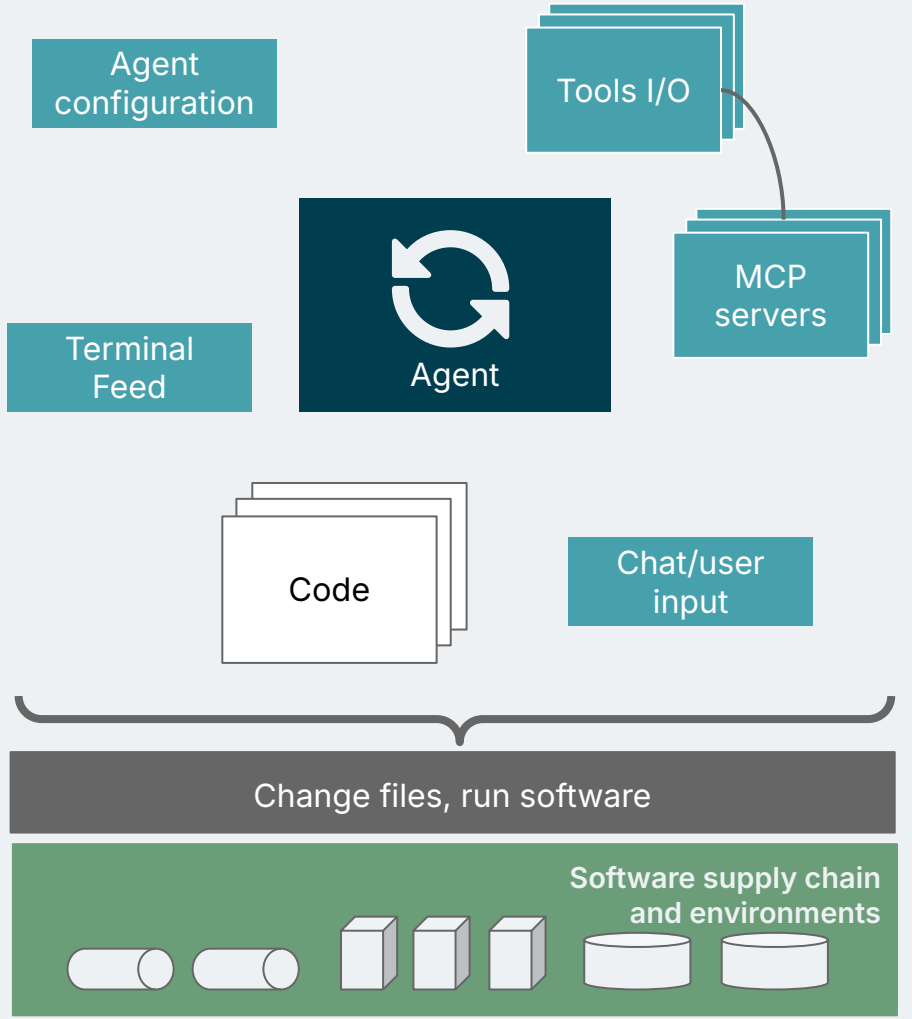
© 2025 Thoughtworks | Confidential

#2. AI Age attack vectors

/thoughtworks

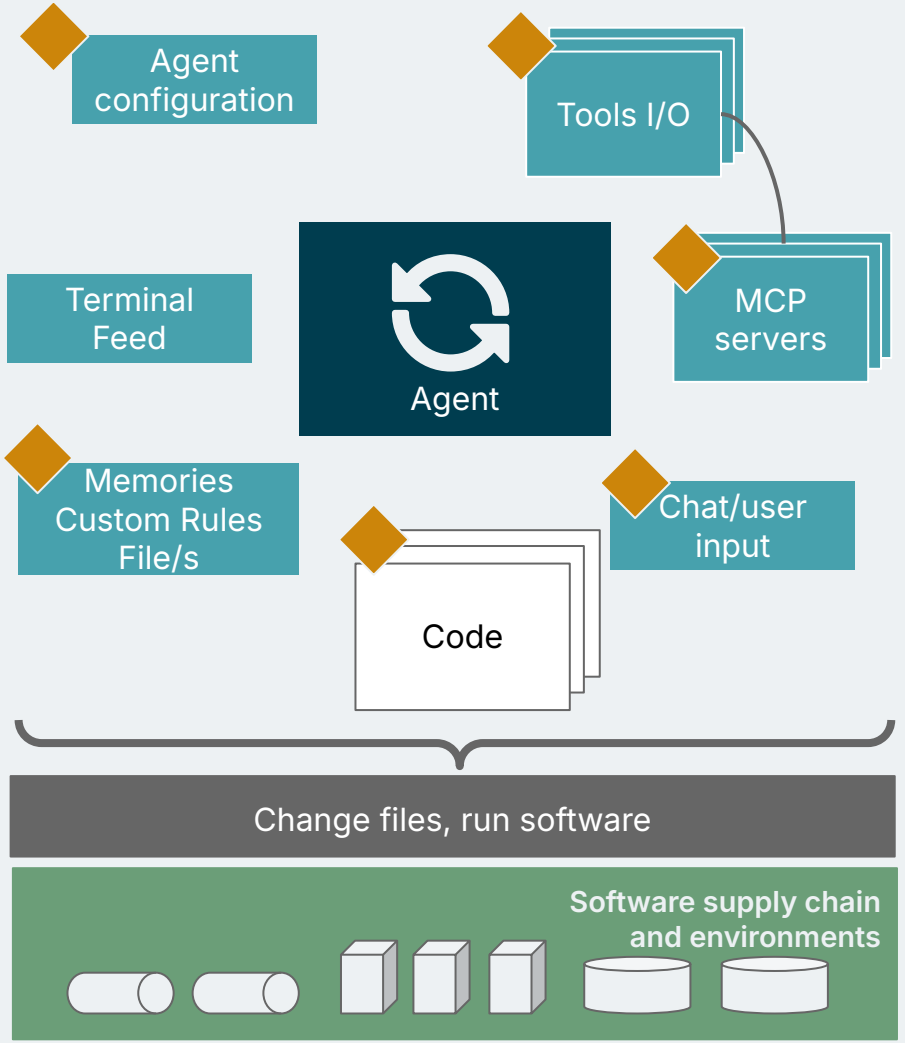
© 2025 Thoughtworks, Inc. Commercial in Confidence

Agentic coding assistant



© 2025 Thoughtworks | Confidential

The “Lethal Trifecta”



- Context poisoning

© 2025 Thoughtworks | Confidential

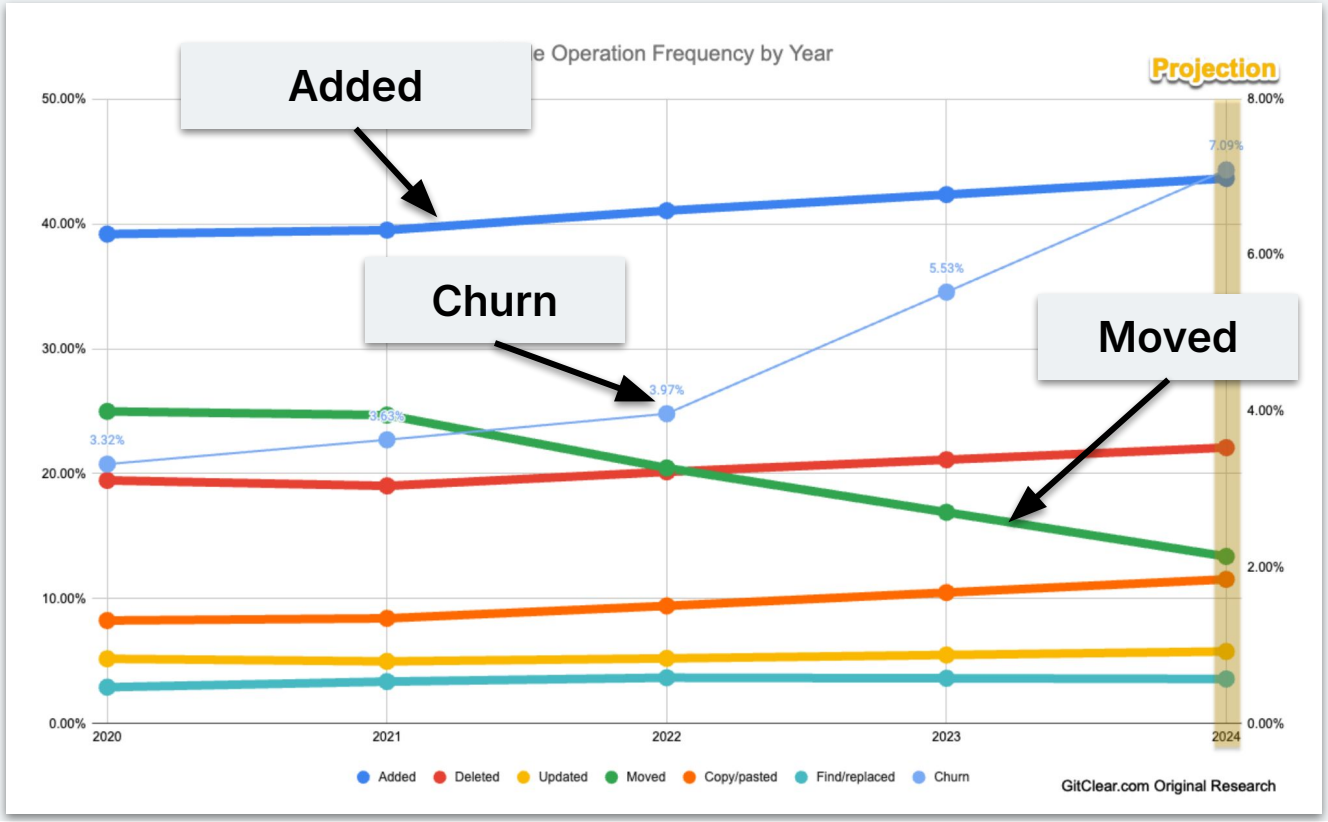
#3. Adverse impact on code quality

/thoughtworks

© 2025 Thoughtworks, Inc. Commercial in Confidence

Are AI-assisted codebases getting bigger faster?

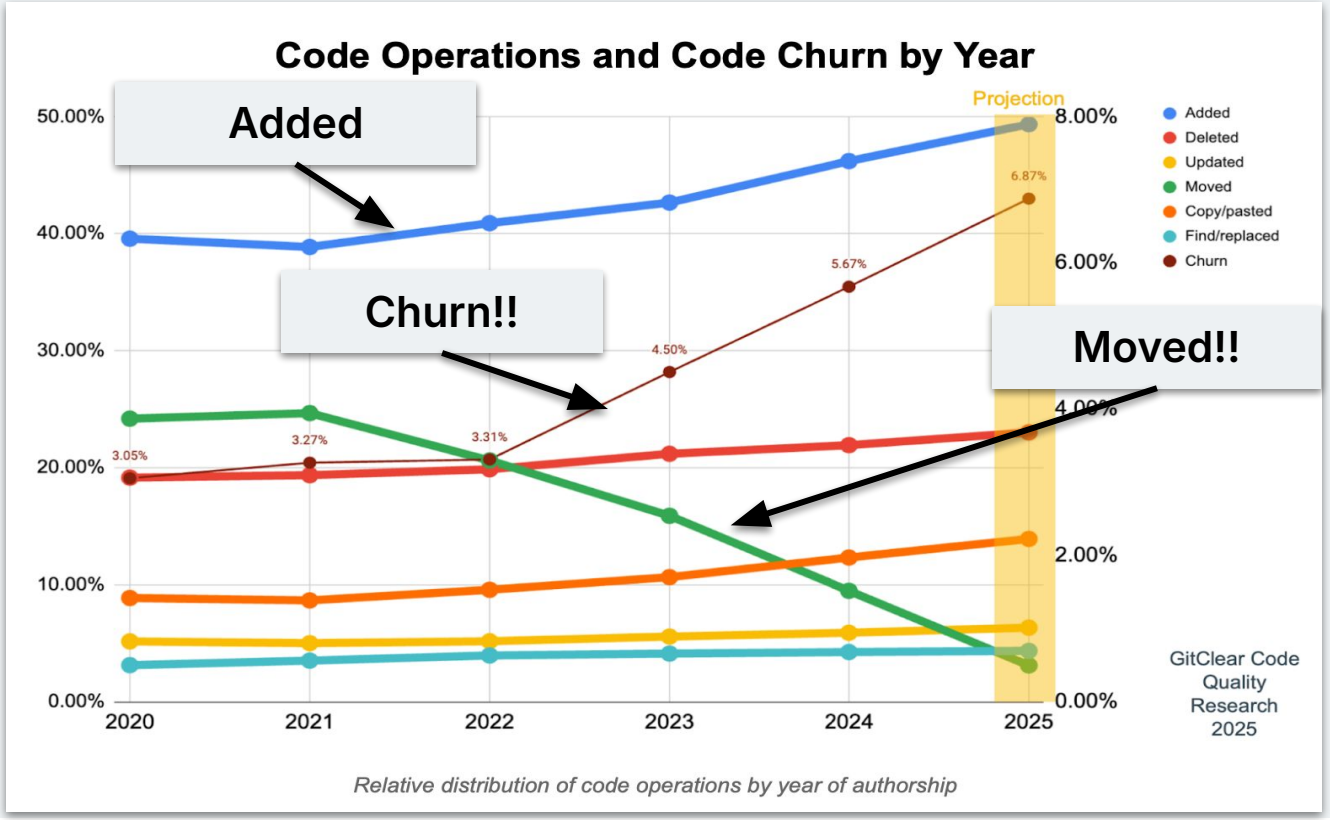
2023



GitClear

Are AI-assisted codebases getting bigger faster?

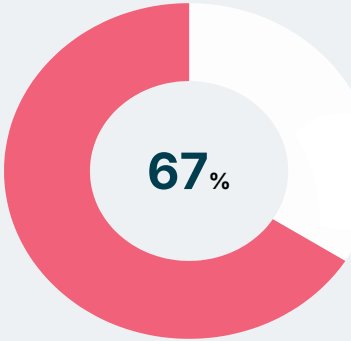
2024



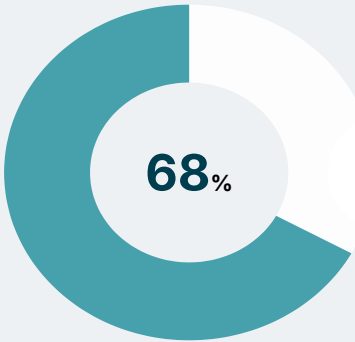
- ↑ Duplicate code
- ↑ Corrections within 2 weeks
- ↓ Refactoring

GitClear

© 2025 Thoughtworks | Confidential



of developers spend more time debugging AI generated code



of developers spend more time resolving security vulnerabilities

92% of developers stated that while AI tools increase the volume of code shipped into production, it also increases the "blast radius" from bad deployments

Credit: State of software development report 2025 by Harness

© 2025 Thoughtworks | Confidential

AI induced complacency is real

TLDR; Don’t become the person who drives into the lake because their navigation system tells them to.



Automation bias

Anchoring bias

Sunk cost fallacy

Security vigilance

© 2025 Thoughtworks | Confidential

#4. Governance and compliance

/thoughtworks

© 2025 Thoughtworks, Inc. Commercial in Confidence

What’s hurting?



Model hosting options - Region, Model as a service,
Self hosting

What's not visible?



Coding model hosting options - Model as service, Self hosting



Lack of visibility on cost incurred and token economics

What’s happening?



Coding model hosting options - Model as service, Self hosting



Lack of visibility on cost incurred and token economics



Rogue AI usage? Enterprise wide usage patterns

What's hurting?



Coding model hosting options - Model as service, Self hosting



Cost incurred and token economics



Rogue AI usage? Enterprise wide usage patterns



ROI on the investment, impact on the engineering effectiveness

Recommendations



/thoughtworks

© 2025 Thoughtworks | Confidential

Recommendations

1.

AI across the entire delivery cycle

© 2025 Thoughtworks | Confidential

AI should boost the entire delivery cycle

1



© 2025 Thoughtworks | Confidential

Engineering practices and craft matter more than ever

1



GenAI tooling can help write code faster, and improve the developer experience.

But the mileage varies based on the circumstances, and experience still matters to uphold stability and quality at the same time.



GenAI tooling should boost the whole delivery cycle, not just coding.

Coding is only one element of the overall software delivery process, and it is not the one that takes the most time. Further opportunities to reduce waste and friction with GenAI exist across the end-to-end lifecycle from feature definition to deployment.



Good engineering practices matter more than ever.

Engineering discipline not only helps mitigate the quality risks of generated code, but is also crucial to deal with the increased throughput. It is also important because GenAI tools amplify what you already have - **it can make the good better, and the bad even worse.**

© 2025 Thoughtworks | Confidential

Super powers of Gen AI

1

How does that help us in software delivery?

Translation

Translating one form of language or code to another allows for many forms of code generation and transformation.

Examples

Requirements to code: Translating natural language to code

Code to code: Migrating code between two systems or tech stacks

Language to queries: Data exploration with natural language

Standard to standard: Translating one standard or API into another

Finding information just-in-time

Chatbots and generative tools integrated directly into user toolchains provide information just in time, exactly for the current context.

Examples

Remembering details and learning things: “How does this work again?”

Understanding errors: Save time when debugging standard errors (“What does this error message mean?”)

Providing organisational context: The right combination of tools can help surface important organisational information to the user right when they need it.

Amplifying and socialising knowledge within a team

Brainstorming and ideation

Chatbots are good brainstorming partners. We can ideate with them, and they can ask us questions and point us to gaps in our thinking earlier in the cycle.

Examples

Product ideation: Generating scenarios and ideas for new features

Requirements: Writing more complete user stories

Testing: Finding gaps in testing scenarios

Architecture: Identifying open questions about architectures

Exploratory testing: “What would happen if...” - let an AI-powered agent simulate a user

Summarisation and clustering

Large Language Models (LLMs) are good at summarising, clustering or rephrasing information.

Examples

Change logs: Summarising commit logs for stakeholders

Incident management: Creating better run books based on incident and deployment history

Research: Summarising and analysing research data to improve the product

Documentation: Technical writing support, or replacing parts of the permanent documentation with LLM-backed search and summary

© 2025 Thoughtworks | Confidential

Our recommendation

1.

AI across the entire delivery cycle

2.

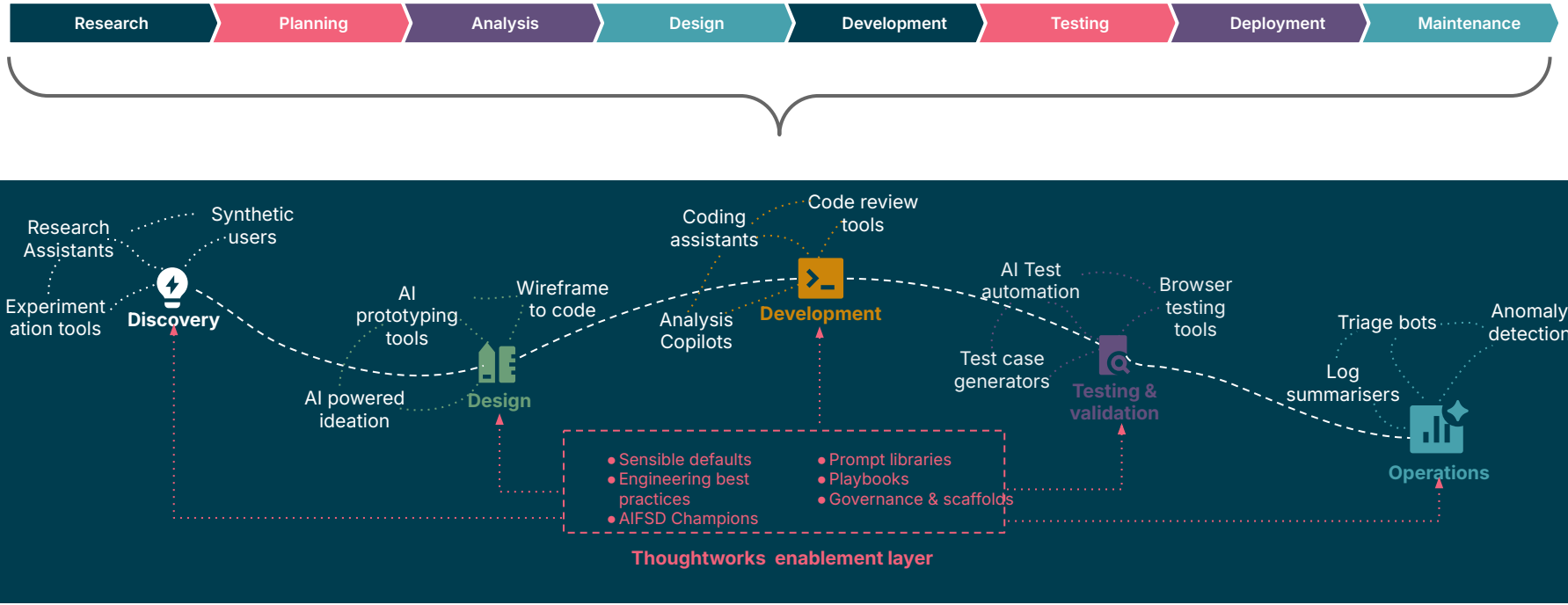
Scale AI across the org with proven workflows via Agentic Platform

© 2025 Thoughtworks | Confidential

Workflow: The missing link in AI Adoption

2

Automate these workflows via Agentic Delivery Platform

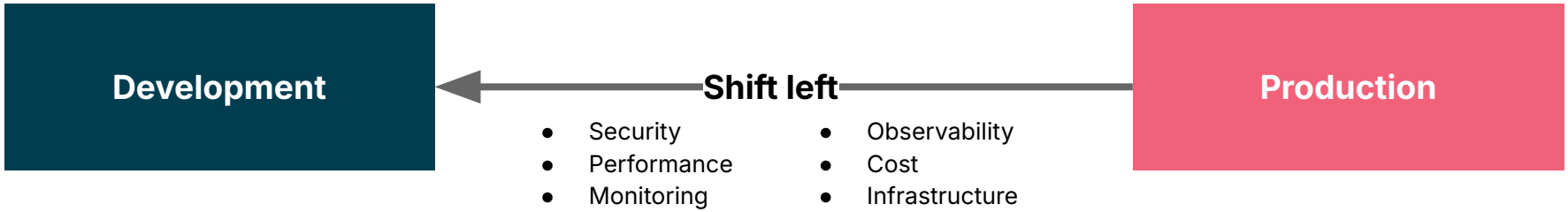


© 2025 Thoughtworks | Confidential

Continuous shift left

2

The need to bring platform thinking from production to development

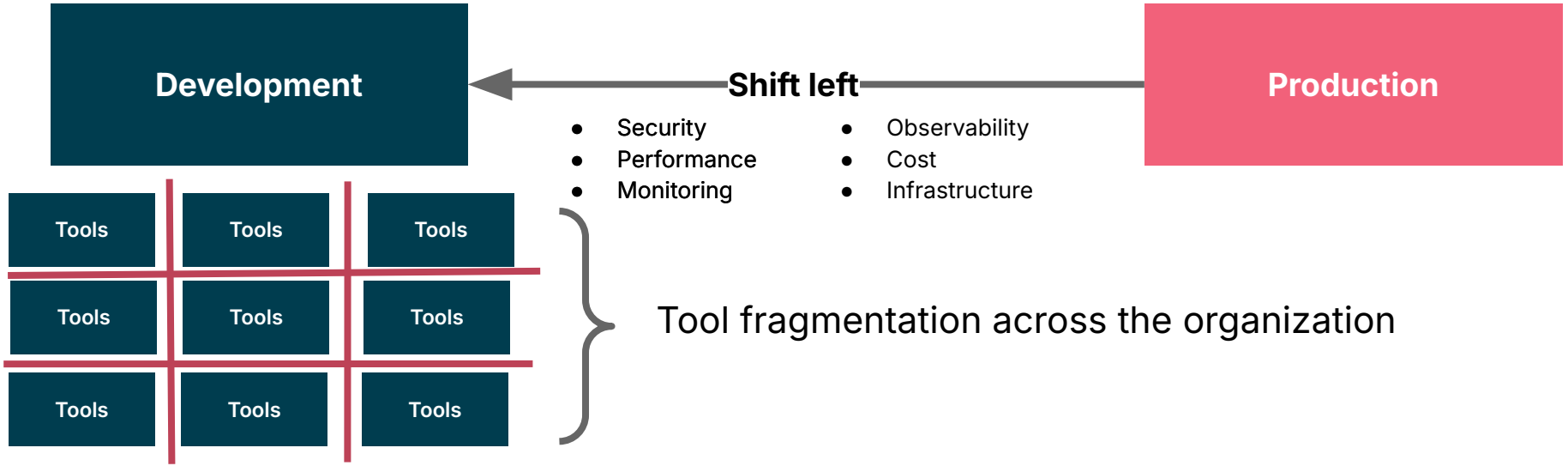


© 2025 Thoughtworks | Confidential

Continuous shift left

2

The need to bring platform thinking to development

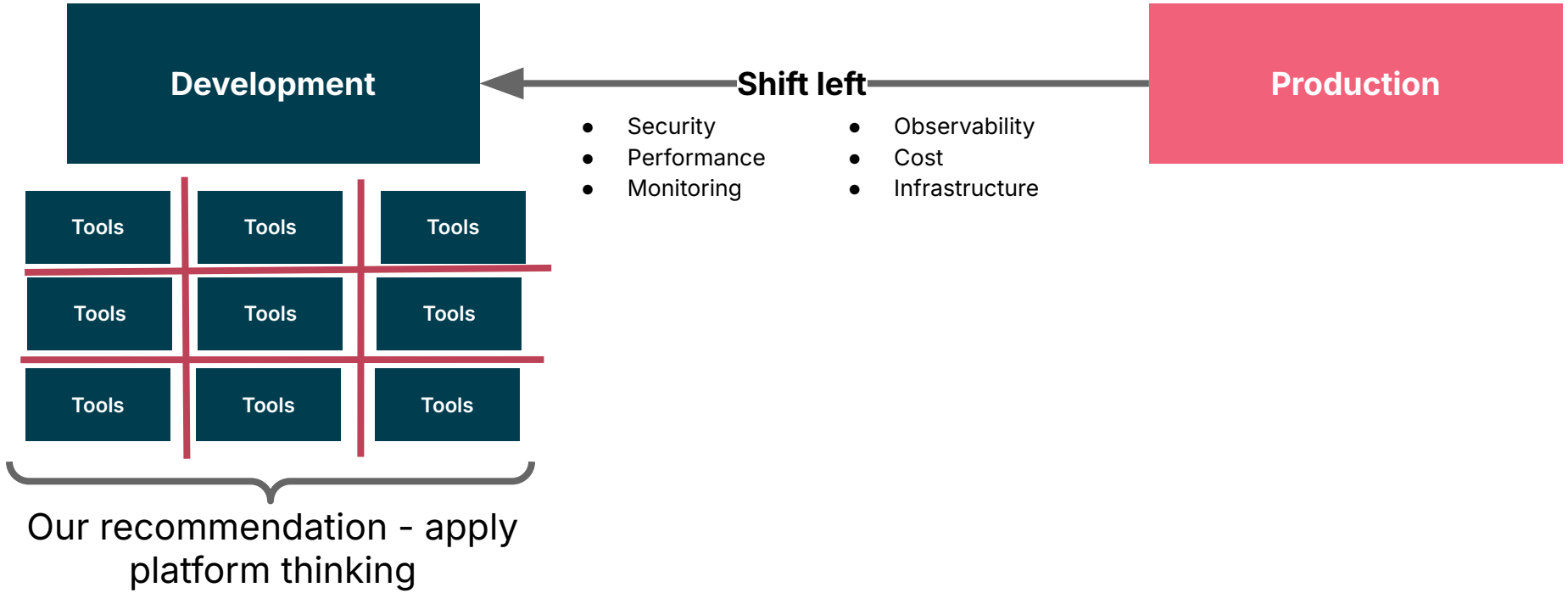


© 2025 Thoughtworks | Confidential

Continuous shift left

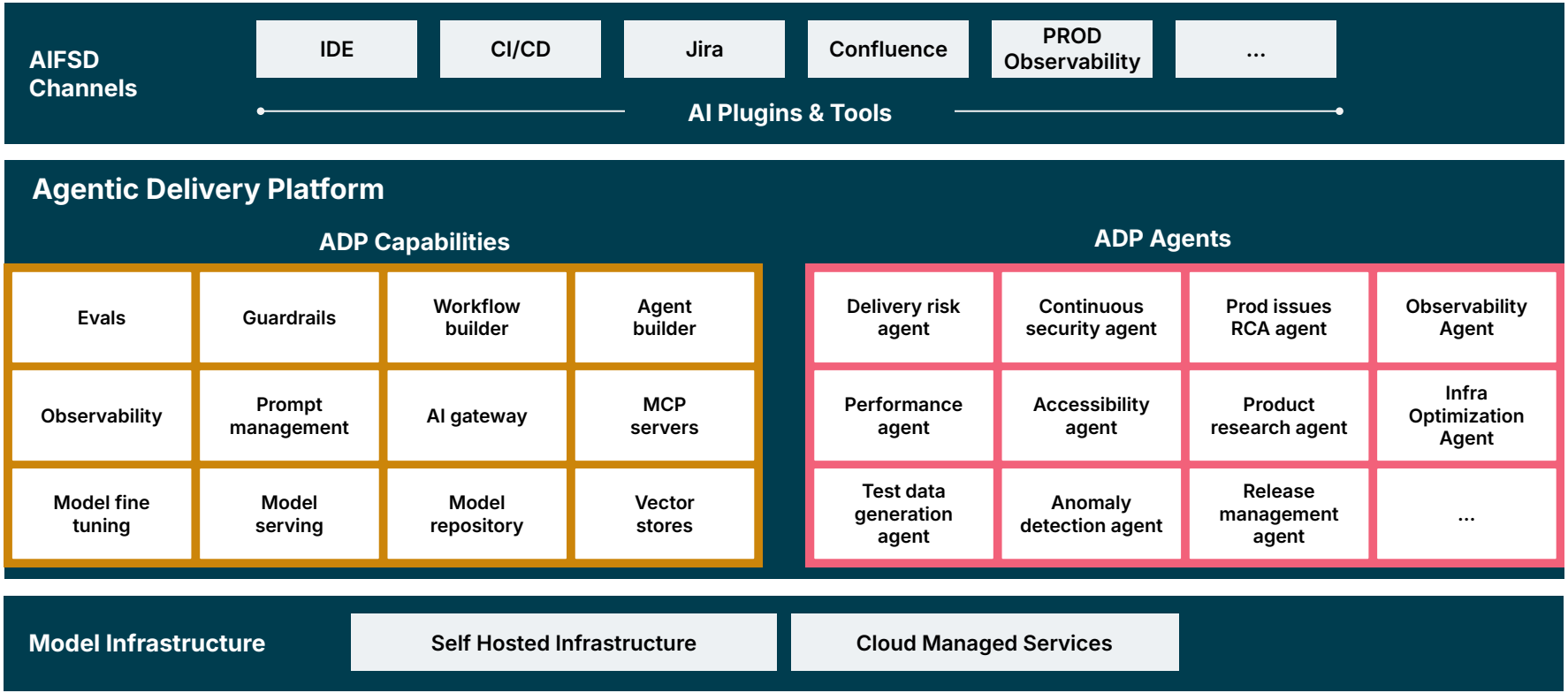
2

The need to bring platform thinking to development

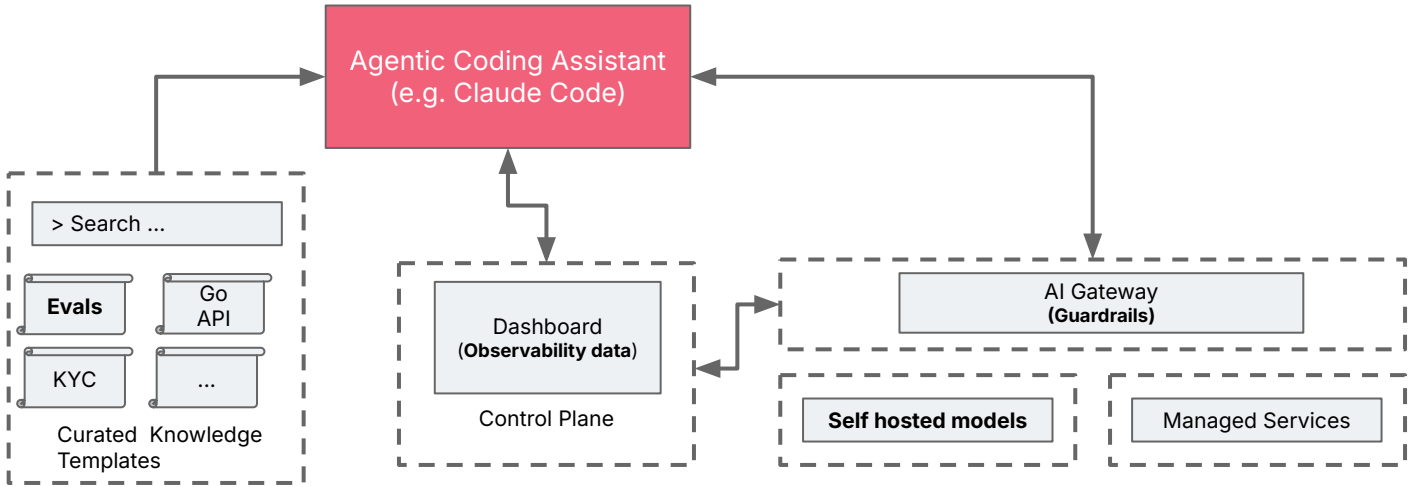


© 2025 Thoughtworks | Confidential

Agentic Delivery Platform
Logical architecture

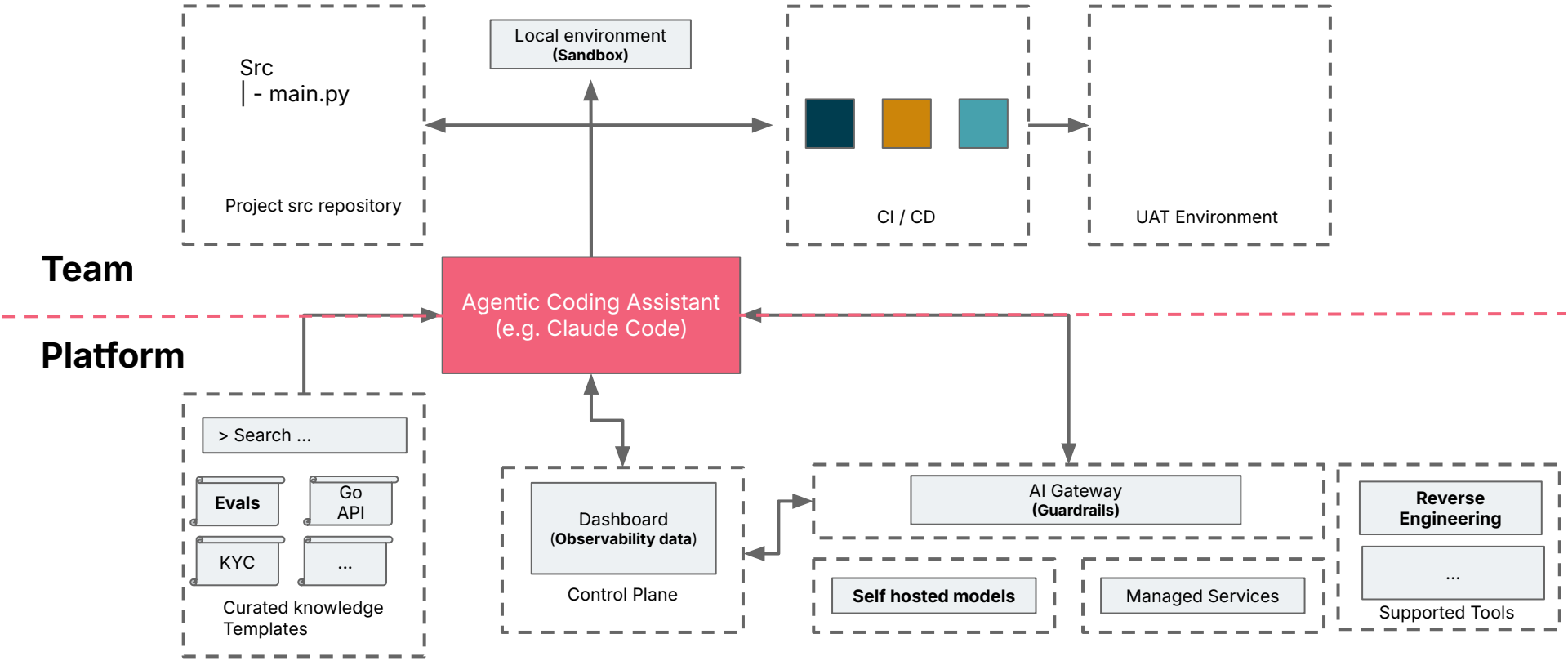


© 2025 Thoughtworks | Confidential

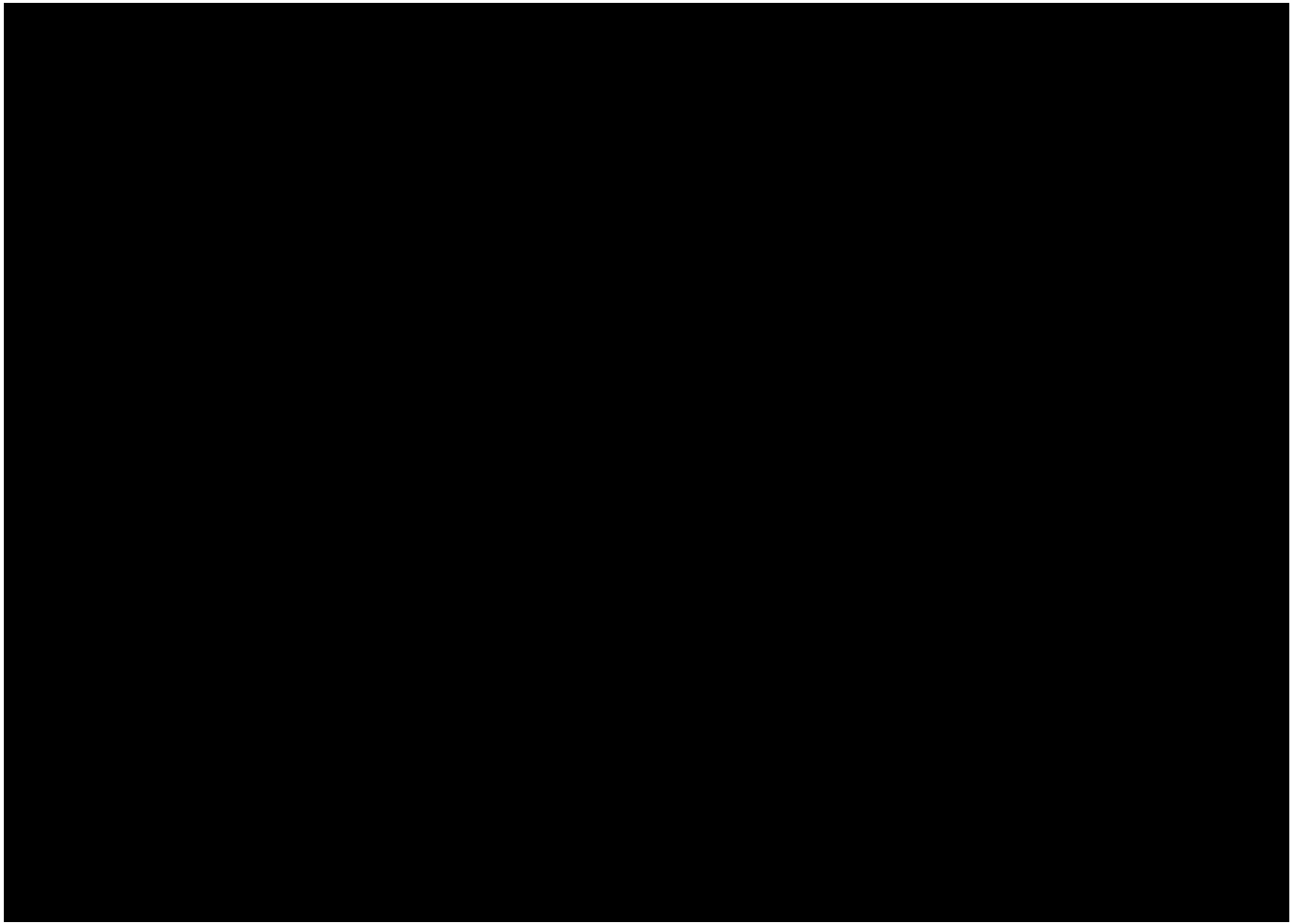


© 2025 Thoughtworks | Confidential

Exemplar developer workflow



© 2025 Thoughtworks | Confidential



© 2025 Thoughtworks | Confidential

Our recommendation

1.

AI across the entire delivery cycle

2.

Scale AI across the org with proven workflows via Agentic Platform

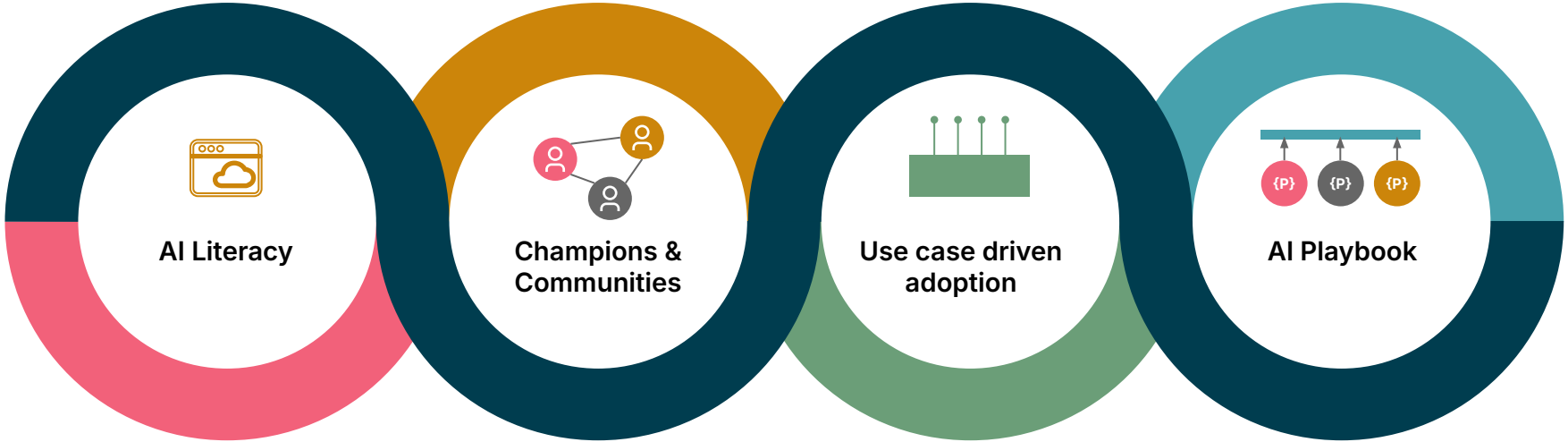
3.

Getting the “crowd”, the people involved in this change

© 2025 Thoughtworks | Confidential

People centricity for this mindset shift

3



Tapping into organizational knowledge to empower all roles

Establishing CoPs to support cross pollination and learning

Teams identify what are their AI use cases based on the superpowers

We capture what's working, what's not working in a playbook as a community driven org resource

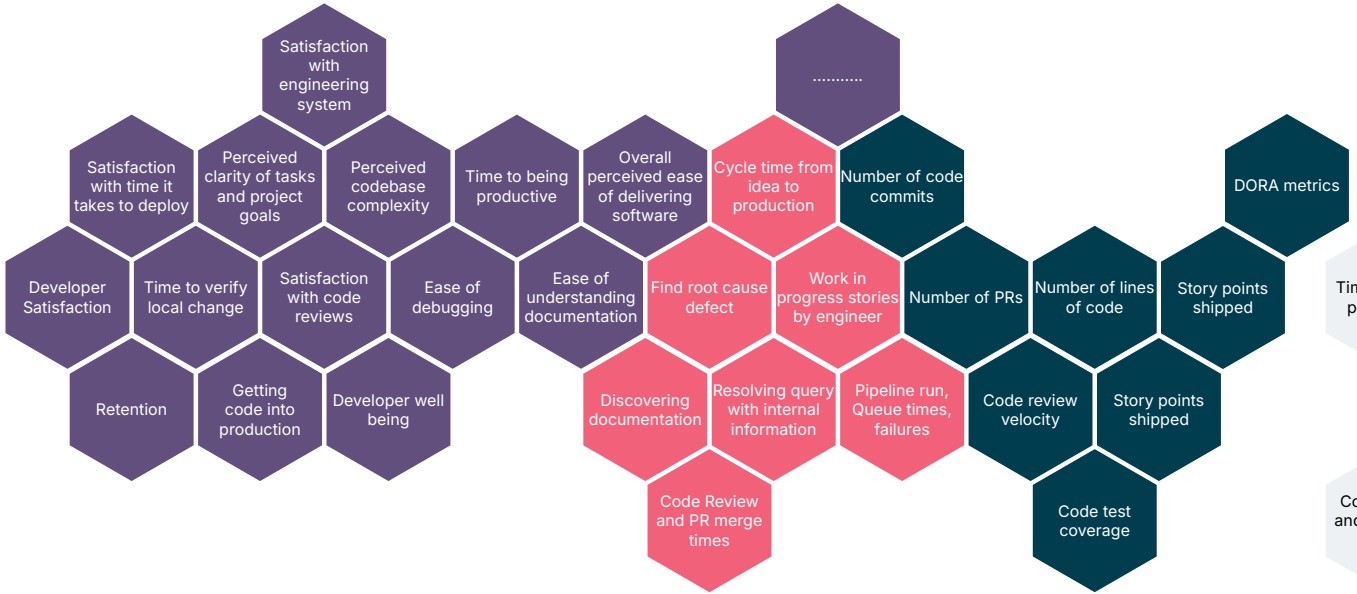
© 2025 Thoughtworks | Confidential

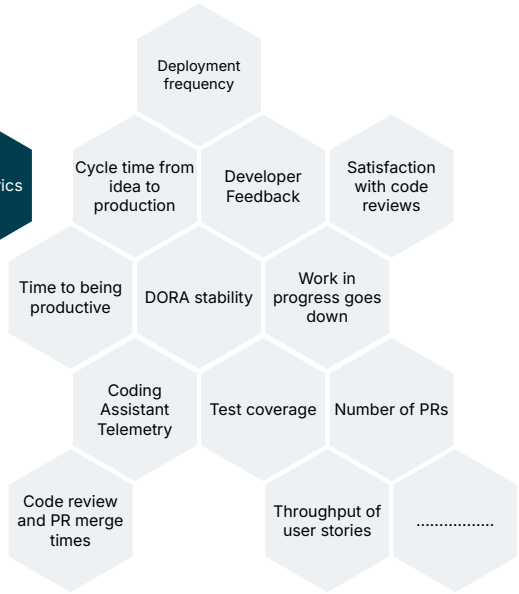
Metrics that matter

3

Spectrum of metrics, no one Metric

What's trending directionally?





Human attitude and opinions ···· **System & Process Behavior** ··· **Activity based** ····· **Are we moving in the right direction?**

© 2025 Thoughtworks | Confidential

Thanks

Vanya Seth
Regional Head of Technology
vanyas@thoughtworks.com

/thoughtworks

© 2025 Thoughtworks | Confidential