AI Boost · October 3, 2025



AI in SDLC: from assistants to agents

Copilots, MCP/A2A, AlphaEvolve, SE 3.0 and large fintech cases

Alexander Polomodov

Speaker About

Technical Director & Fellow, large fintech

- Architecture and engineering practices in fintech
- AI in SDLC across thousands of engineers
- Podcast «Code of Leadership», @book_cube

Agenda Plan

From tools to measurable outcomes

Why agents emerged, how to build them, and how to measure their effect

- Why agents became possible
- How engineering work changes
- How agents enter the SDLC
- How impact differs from usage

Part 1

01

Previously on this show

How we integrate AI into SDLC — big tech cases

Recap Big Tech AI

Agents continue the tooling evolution

The previous talk covered AI integration; this one follows the autonomous loop

- Assistants already live in tools
- Autonomy is the next step
- Agents need a shared platform
- Outcomes still need proof

Part 2

02

Assistants → agents

From autocomplete to autonomous task execution

Assistants became everyday tooling

Tools | Assistants

Copilot and Cursor opened the path; Claude Code and Codex pushed agent mode

- Copilot: agent mode
- Cursor: AI-centered IDE
- Claude Code: terminal agent
- Codex: long-running tasks

An agent owns the task loop

State of the art Now

The human states the outcome and decides how deeply to inspect the result

- Plans its own steps
- Calls external tools
- Acts without step-by-step commands
- Returns a result to the human

Part 3

03

Protocols: MCP and A2A

Standards for agents and integrations

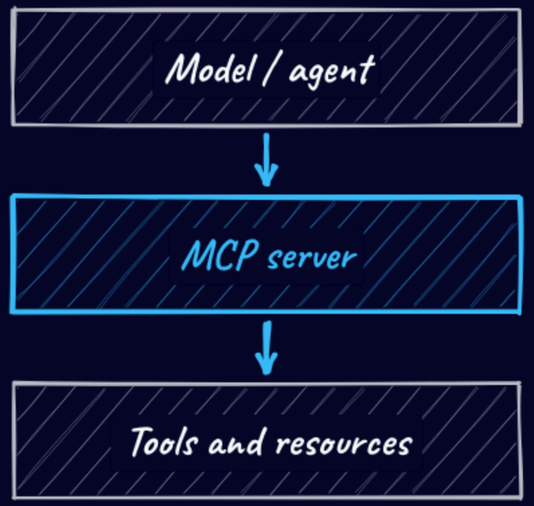
MCP connects tools; A2A connects agents

Protocol MCP · A2A

MCP

Model Context Protocol

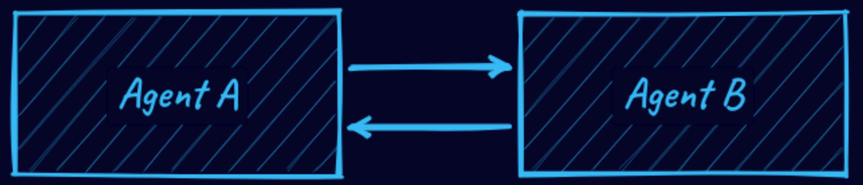
Model context and tools



A2A

Agent to Agent Protocol

Communication between agents



Shared rules and tasks

Part 4

04

Research and economics

Why agents are real and here to stay

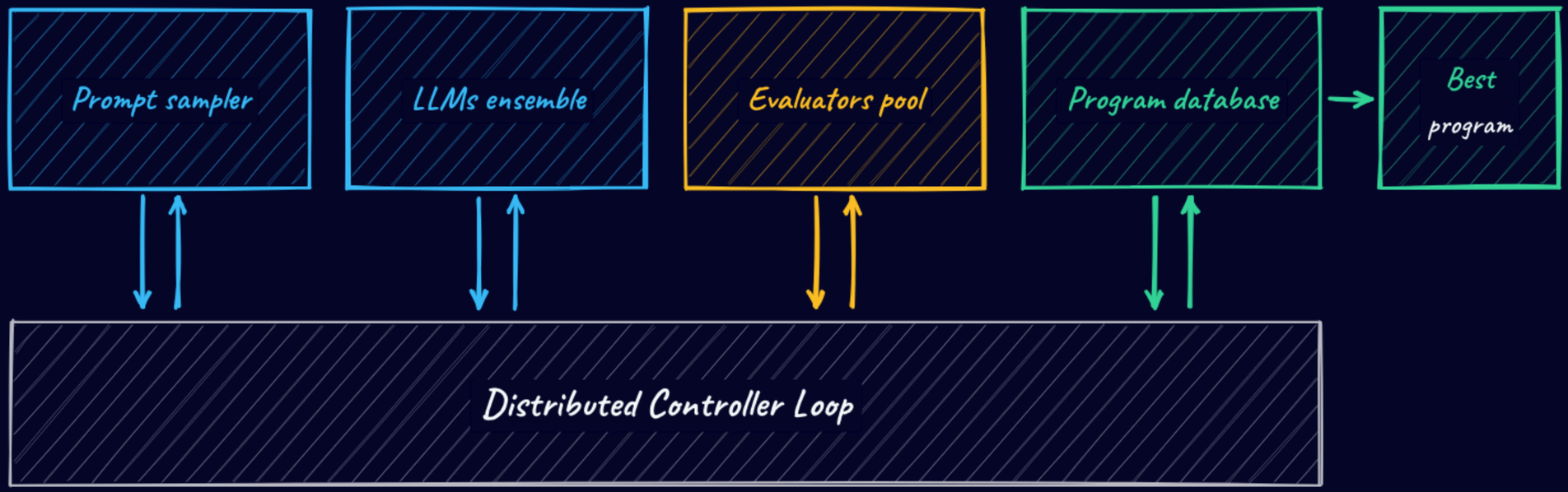
Three forces push the industry toward agents

Investment demands returns, business demands efficiency, and the market expects promised outcomes

- Billions invested since 2023
- Pressure on company efficiency
- Promises from AI providers

AlphaEvolve improves programs through evaluation

Research DeepMind



Research AlphaEvolve

A measurable evaluator turns search into impact

Google applied one loop to infrastructure, hardware, and model training

- Borg scheduling: +0.7%
- TPU circuit optimization
- Faster matrix computation
- Gemini training: +1%

People do not want every feasible automation

Research Stanford

Compare worker desire with the actual capability of current agents

Green light	R&D	Red light
Wanted by workers, feasible today	Wanted by workers, not feasible yet	Feasible today, unwanted by workers

Agent economies need deliberate rules

Research Google

The authors prefer designed sandboxes over an emergent, permeable market

Low risk

- Permeable sandbox
- Rules designed up front

High risk

- Bounded sandbox
- Rules designed up front

Part 5

05

SE 3.0 — AI-Native Engineering

A new paradigm of software development

Paradigm SE 2.0 → SE 3.0

SE 3.0 moves the interface from code to intent

An AI-native process connects human requirements with machine implementation

- SE 1.0: code-first + tools
- SE 2.0: code-first + AI
- SE 3.0: intent-first + dialogue
- Humans define requirements

Start with the developer goal, then choose tools

Google research DevEx

The developer journey links SDLC stages, durable goals, and platform teams

- Gather information
- Plan and gain approvals
- Develop, test, and commit
- Release and operate

Part 6

06

Large fintech case

Spirit and agent mode in production

Agent-first starts with the whole platform

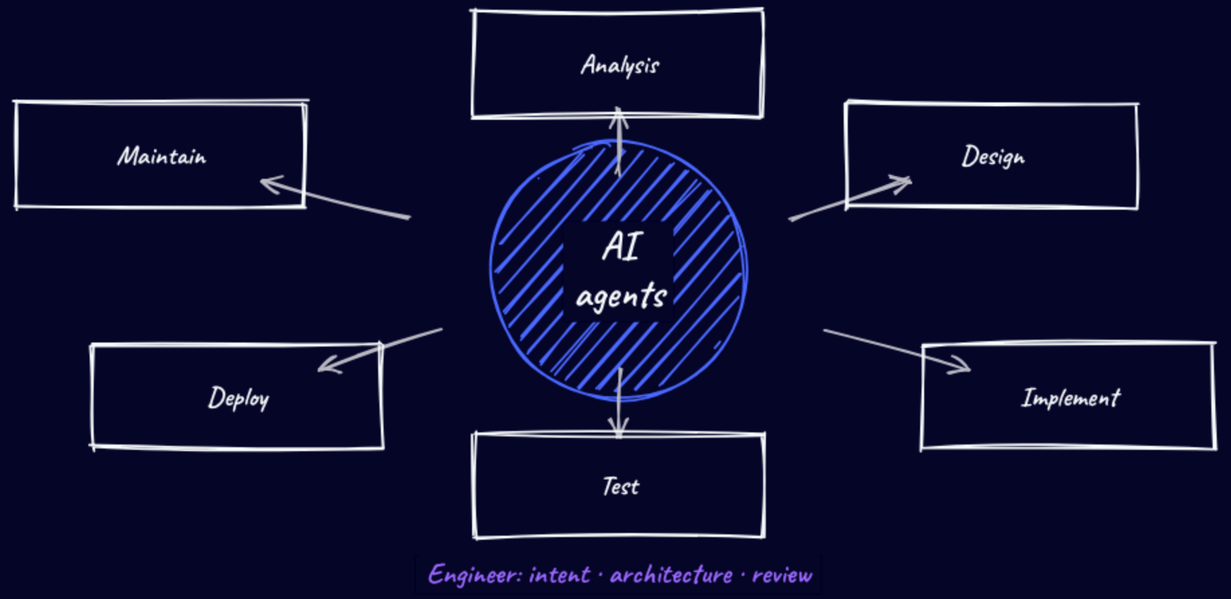
Platform Spirit · Internal PaaS

Spirit spans work intake and knowledge through infrastructure and production readiness

- Process, search, and documentation
- VCS, IDE, CI, and builds
- Delivery, cloud, and components
- Observability and incident management

Agents must cover the whole SDLC

The engineer sets direction, designs the system, and reviews the output



Reality check | Time split

60% of time is not code writing

Web and mobile team analysis: an AI program cannot stop at autocomplete

Creation

- Code — ~40%
- Tests — ~15%

Coordination

- MR / code review — ~15%
- Jira — ~12%
- Wiki — ~12%

Agents move beyond the IDE

Agent mode Professions

Teams choose painful jobs across the lifecycle

- Code: complete a task end to end
- Data: request → code → chart
- QA: description → test case
- Each profession chooses its jobs

Part 7

07

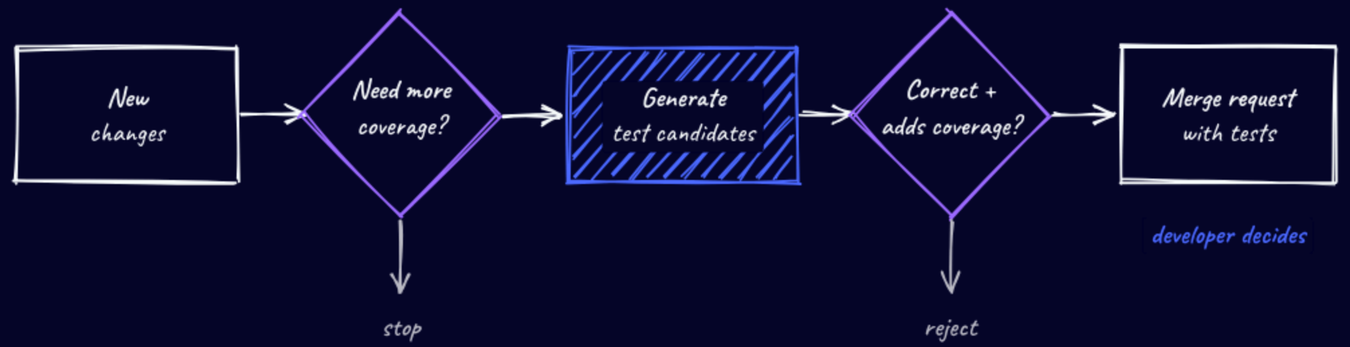
Agents in production

T-Cover Agent, AI code review, security

Testing T-Cover Agent

T-Cover proposes only validated tests

Change → coverage need → candidates → validation → MR for a developer decision



Testing T-Cover uplift

Lower baseline coverage yields larger gains

Added coverage declines as the existing test base grows

Up to 66%

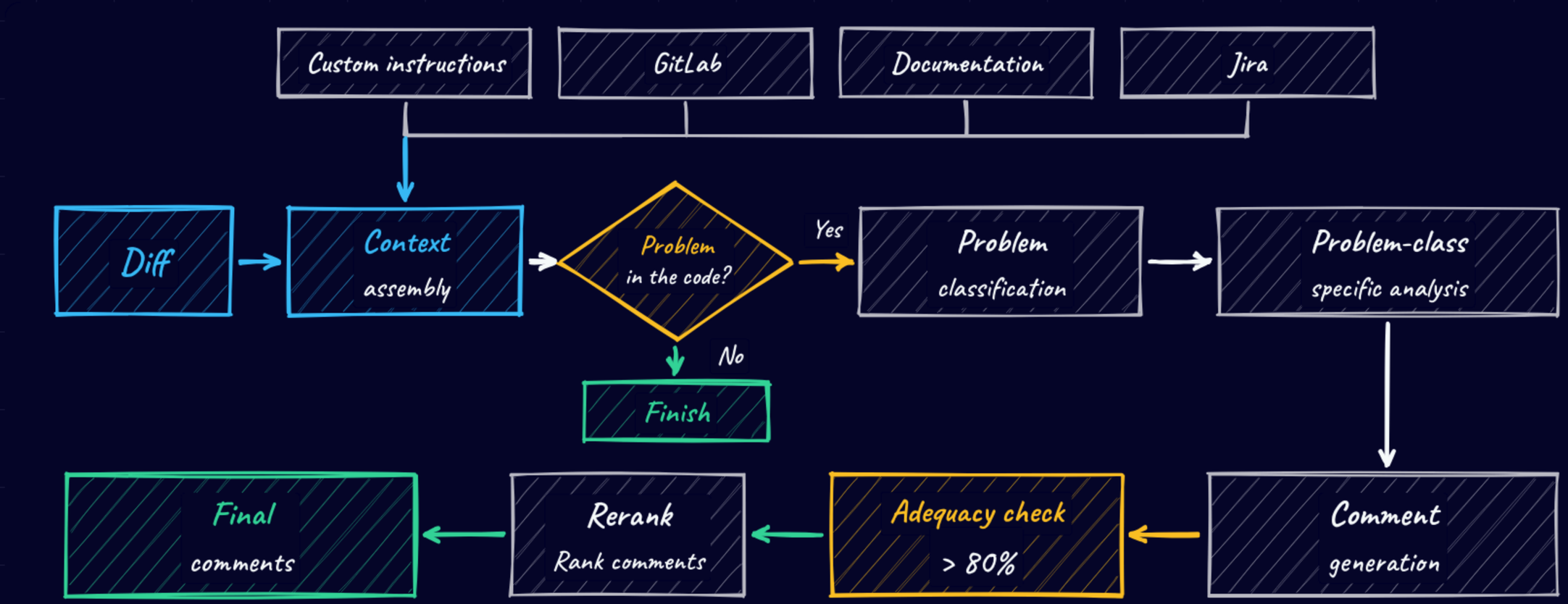
- 0–20% → +40 pp
- 21–47% → +33 pp
- 47–66% → +27 pp

Above 66%

- 66–73% → +23 pp
- 73–89% → +12 pp

AI review starts with context

Code review | MR pipeline



A security agent sees code, not every control

Security InfoSec

SafeLiner finds a vulnerability and proposes a fix, but may miss infrastructure mitigations

- Scans the repository
- Explains the finding
- Offers a one-click patch
- May miss infrastructure protection

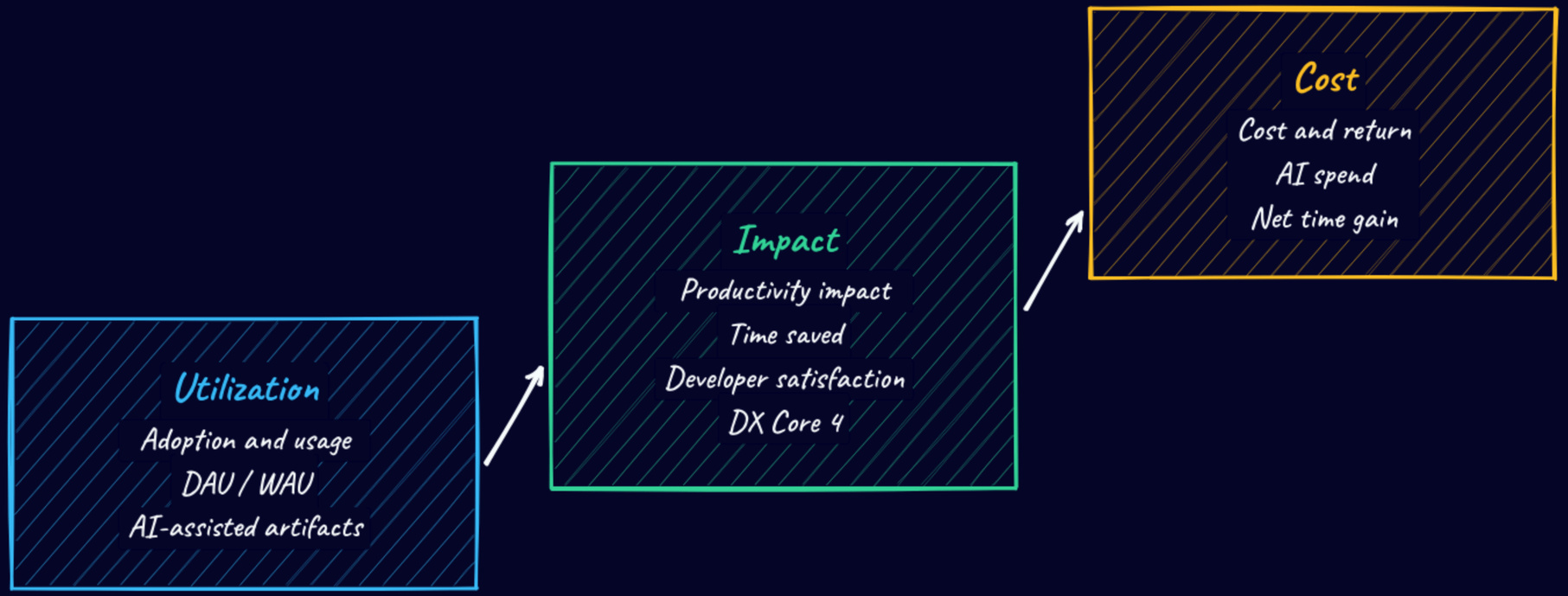
Part 8

08

Measuring impact

Usage is only the first level of evidence

Utilization starts; impact still needs proof



Impact T-Bank · 2025

Usage grew; impact still needs proof

Internal data clearly establishes adoption and retention

Created artifacts

- IDE H1: 10%
- IDE H2: 25%
- Heli: 5% → 10%

Retention · 4 weeks

- IDE: suggest 79%, chat 55%
- Heli: suggest 70%, chat 46%

Research AI4SDLC Russia

The Russian market needs a comparable baseline

The study connects practices, engineer perception, and measurable outcomes

- Practices across Russian companies
- Quality and efficiency
- How engineers perceive AI
- Comparison with global trends

↗ ai4sdlc.tbank.ru

Agentic development is a system

Five conclusions from research and production cases

- Protocols connect the ecosystem
- Evaluators turn search into outcomes
- Platforms span the entire SDLC
- Engineers state intent and review
- Utilization is not impact

Sources References

Sources behind the argument

Research on discovery, work, agent economies, SE 3.0, and measurement

- AlphaEvolve · Google DeepMind
- Future of Work · Stanford
- Virtual Agent Economies · Google
- SE 3.0 · Measuring AI · DX

polomodov.tech

Thank you!

All slides and links are in the Telegram channel

Alexander Polomodov

Technical Director & Fellow,
large fintech



@book_cube