

# Agentic AI SE - Program with Trust

Baishakhi Ray & Abhik Roychoudhury

June 3, 2026

# Many thanks !

IBM for hosting the event !

Maja Vukovic, Xuan Liu, Eugene Wu (Columbia)

# Goal of the sharing

Discussion, Community Building ...

+

Evolution of AI for Code Field

**Enabling greater trust – beyond excitement around capability of coding agents**

Challenges in deployment in production – trust of course, but could be more ...

*Putting agents in production*

*[Do not get into the hype of “coding agents are great” ]*

+

*Putting agents “right” when they eventually go to production e.g. Security issues*

*[Avoid disappointments]*

# Automatic Programming

Agentic AI Software Engineers: Programming with Trust  
Abhik Roychoudhury, Corina Pasareanu, Michael Pradel,  
Baishakhi Ray

Communications of the ACM, 69(5), May 2026.

< and many others ... please add here based on inputs  
from today >



Op

Ag

Pr

Rc

Cc

# Factors for Trust in Coding Agents – CACM Article



What is need for trust?

How can trust be achieved?

Technical

Correctness

Security

Performance

Maintainability

Compliance

Reviewing & Testing

Dependency Analysis

Profiling

Metrics

Static Analysis

Human

Explainability & Transparency

Bias & Ethics

Team Practice

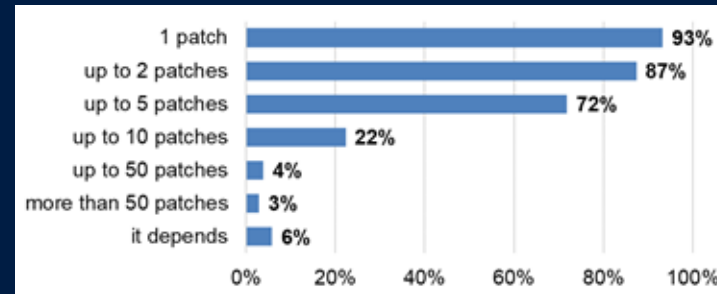
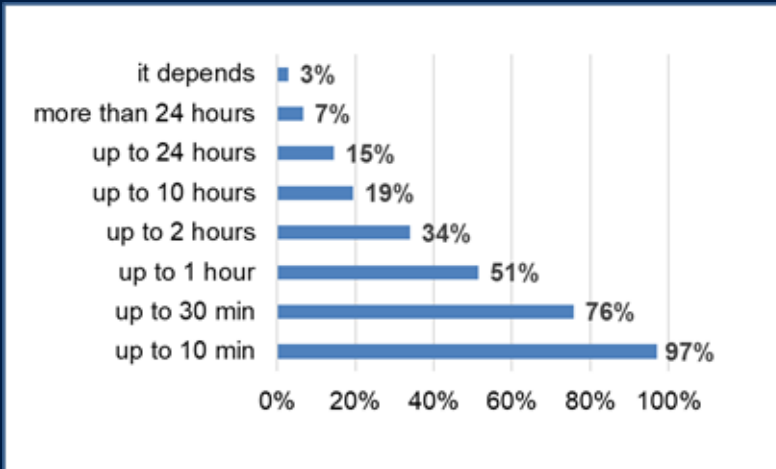
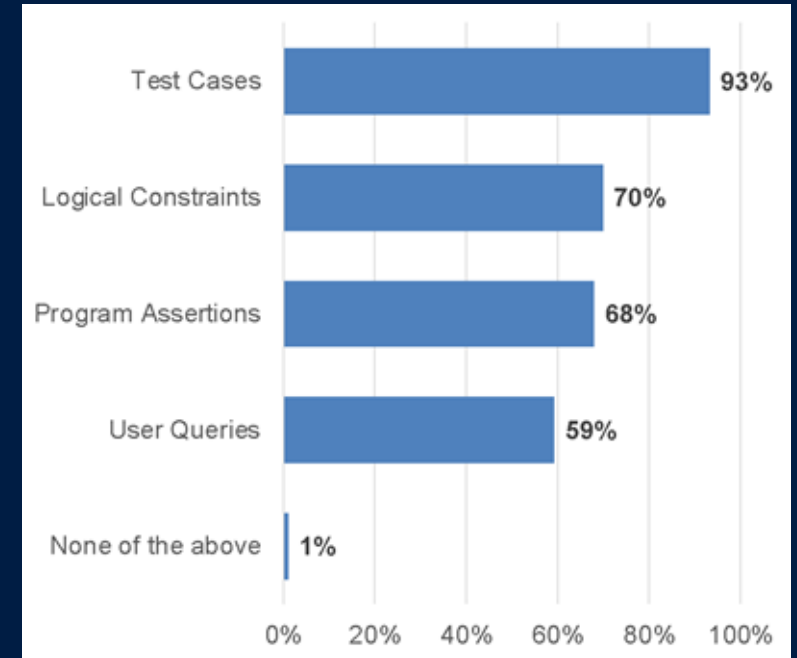
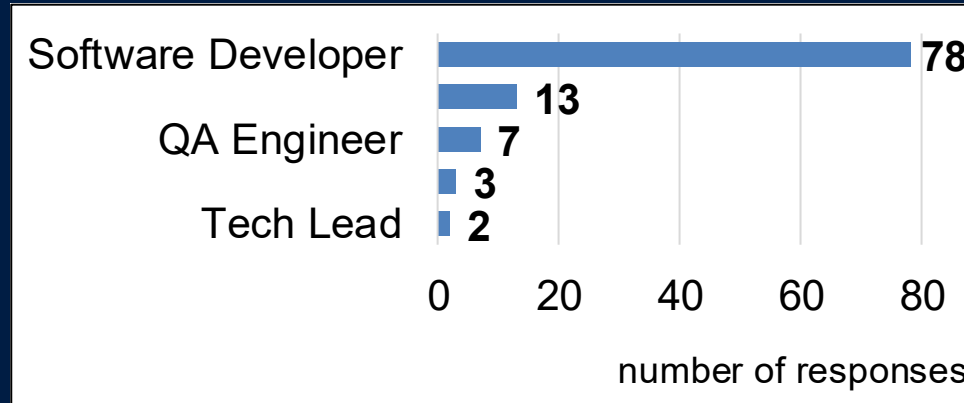
Collaboration

Familiarity

Over-reliance

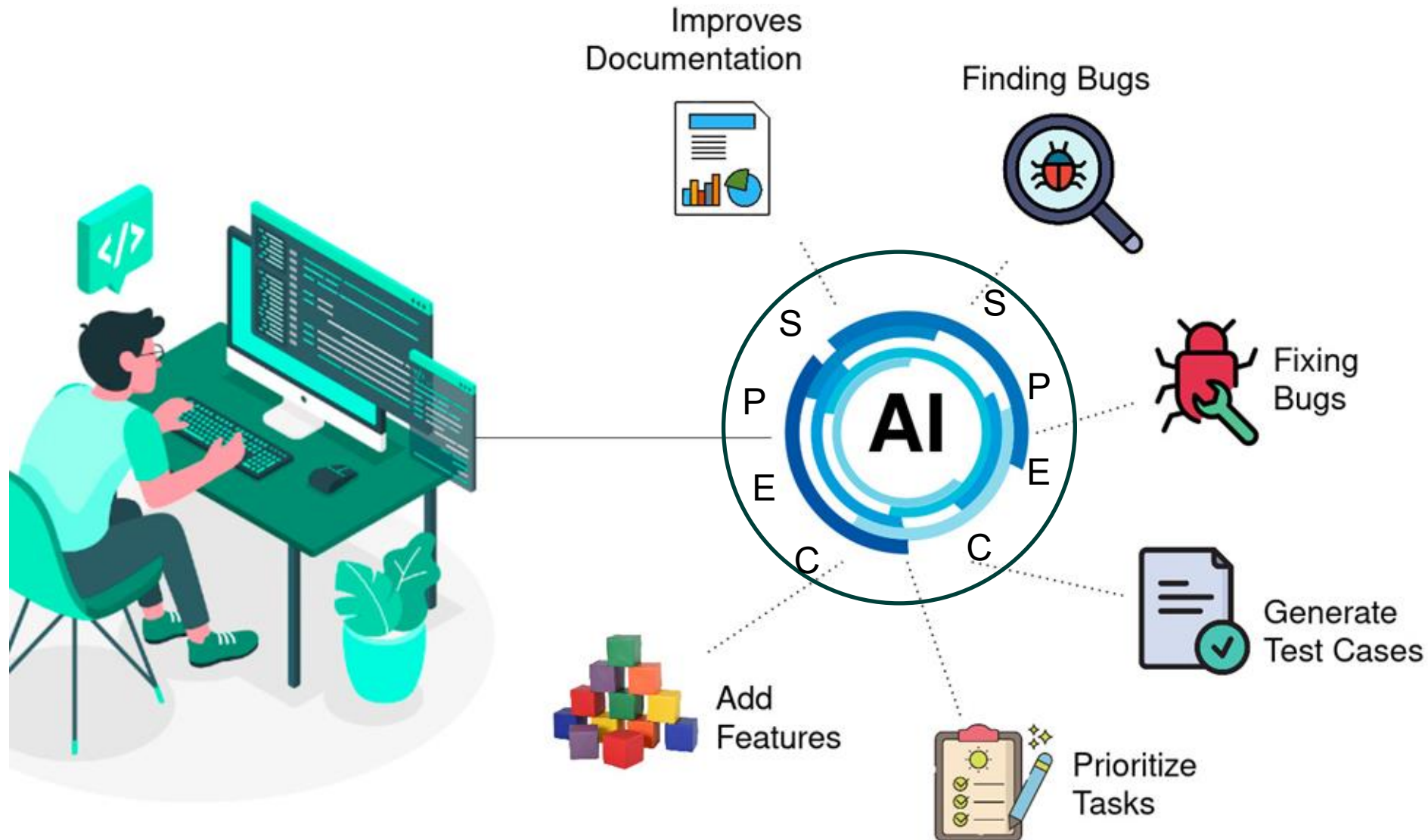
Experience Mismatch

# DEVELOPER'S THINKING ON AUTO-CODE – pre-LLM



Which of the following additional artifacts will increase your trust?

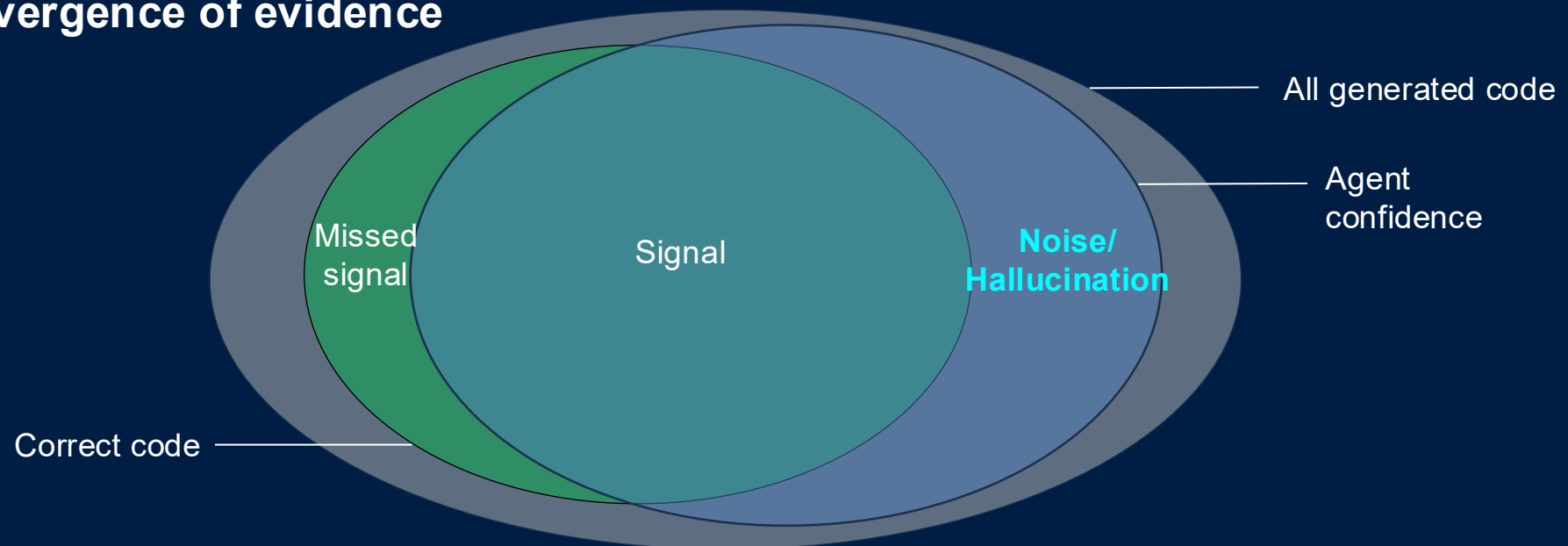
# Autonomous Program Improvement



An AI powered DevAssistant working with the developer to autonomously improve the program

# Beyond Efficacy: Signal-to-Noise Ratio

- Agents generate massive amounts of code
- Agents should be **well-calibrated** (better say no than **hallucinate**):
  - Reduce human review burden
  - Reduce problematic code going into production
- **How to improve? Convergence of evidence**
  - Testing
  - Formal proof
  - Static analysis
  - AI-based review
  - ...



# Agent Explainability – Cognitive Debt

An example of agent explanation from  Claude

- Perfect! I've successfully modified the maze\_game.py file with all the requested features:

Changes made:

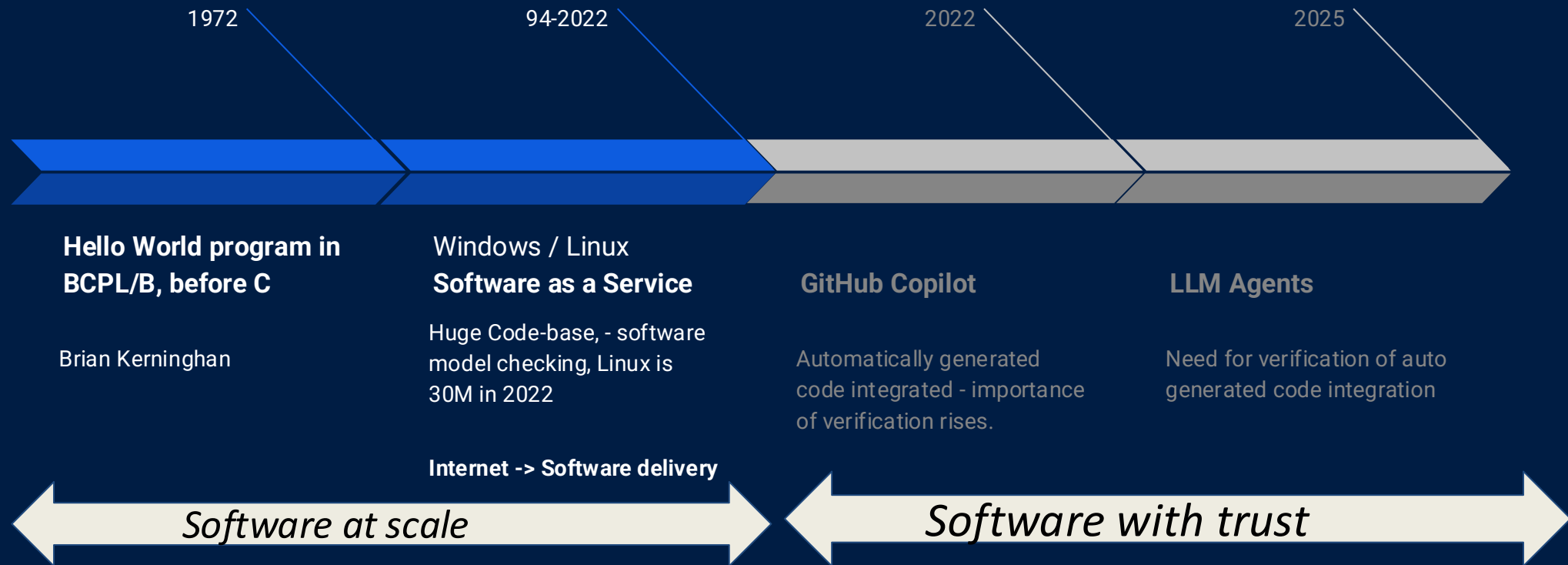
1. Bigger and Harder Maze:
  - Expanded from 9x12 to 15x20 (almost doubled in size)
  - Added more complex paths with additional walls and obstacles
  - Longer route from start to goal position
2. Solve Counter:
  - Added `times_solved` counter that tracks how many times the maze has been completed
  - Displayed in the game header alongside the move counter
3. Maze Reset Feature:
  - Created `reset_maze()` method that resets the maze to its original state
  - When the maze is solved, the game now:
    - Increments the `times_solved` counter
    - Shows completion message with stats
    - Asks if the player wants to play again
    - Resets the maze and allows continuous play if they choose "Y"

The game now supports continuous play sessions where players can solve the maze multiple times k their total completions!

- Users primarily interact with natural-language explanations of what the agent claims to have done.
- This makes agent explanations critical to building trust in AI coding systems
- Important questions:
  - Are such explanations **sufficient**?
  - Are such explanations **trustworthy**?

Can property-based tests be extracted and presented as NL explanations

# Way forward



**Resilience across different dimensions  
Beyond the AI hype --- but well versed with it.**

# Discussion points from SG roundtable – Jan 26



## 1 Improving Trust in Agents

Agent precision is more important than recall

Tools help manage context length

A new language to interact with agents is needed.



## 2 Discussions on agent QA

Productivity increase is difficult to measure, making agent QA difficult

How to test agents and assess their readiness is an important topic



## 3 Future of software engineering

Agent trajectories are important artifacts for SE.

Development might be massively parallel with agents making development easy.

# What is easy to cover

Writing of an agent – reactive loop

Working with coding agents

Automation of SE activities : testing / code review

## Obvious areas to look into

V & V : Holistic view – testing / verification / review/ inspection / analysis / metrics

HCI : Agent human collaboration in a cooperative team – new dynamics

# What is hard to cover

Risks in deploying agents

- But should be covered

Going beyond the efficacy of coding agents

- Explainability - all agents now have explanations
- Signal to Noise ratio

Cognitive debt due to coding agents

- Differentiating from traditional technical debt in SE
- building of tools and processes to tackle cognitive debt in the new SDLC ?

Integration of agent generated code with legacy manual code (discussion)

- any ideas about setting assignments to validate such code integration ?

# Roundtable topics

Future of SE - what new skills are needed in engineers

- New SE processes and workflows
- Role of V&V
- How to train models with trust mindset

# Sub-topics for each group

Future of SE - what new skills are needed in engineers - how to prepare for the future.

- *Job avenues / Curriculum*
- *Change in emphasis of SE tech: testing / verification*
- *Agent human dynamics*
  
- New SE processes and workflows [Rahul Krishna].
  - *CI/CD*
  - *Cognitive Debt management*
  
- Role of V&V
  - *AI-based V&V of AI-generated code*
  - *Holistic view – Testing / Verif / Code Review / Inspection / Code quality metrics*
  - *Program proof co-evolution*
  
- How to train models with trust mindset
  - *model features relevant to secure code gen.*
  - *pre-training vs. fine-tuning*

# What skills are needed in future SE ?

Forward deployment engineer

- Can see the whole process as an organizational workflow evolution
- Which parts to automate, how, and what are the risks ?
- A bit more holistic view than simply which LLM to use?
- **Systematic design and integration of agents in workflows.**

MLSys expertise is useful for some jobs

- core expertise systems, low level SE / PL, hardware, distributed systems

Pre-sales of many AI / AI adjacent organizations involves heavily technical skills

- combine CS technical skills with communications skills.

< ... and many more based on today's discussions ? >