AI Native Software Engineering (AINSE)- What changes when AI Starts Developing Softwares?

Presented by: Sahil Dua, Sunayana Hazarika

What is AINSE?

Scan the QR Code for the executive brief

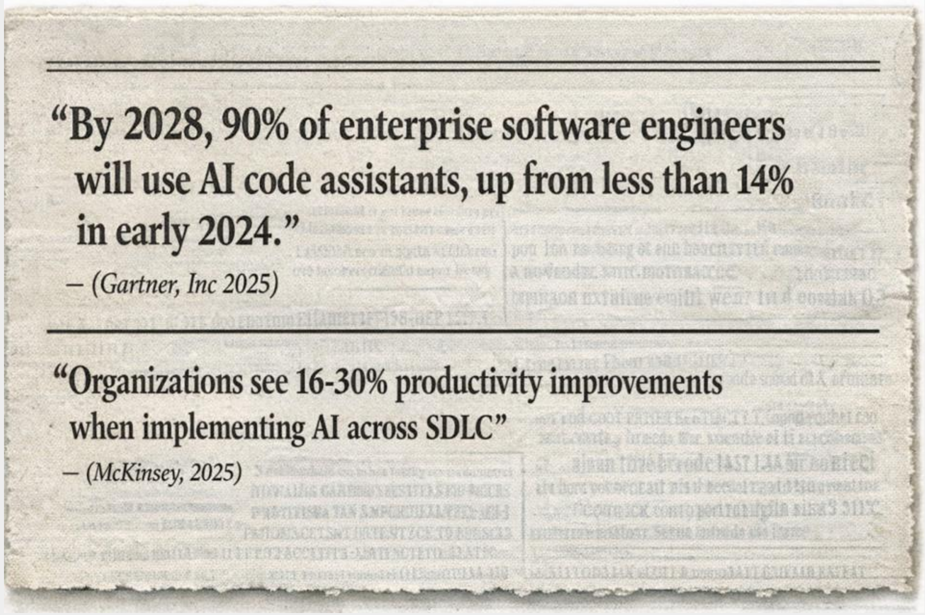
AI in Development?

We are entering an era where software writes software.

Moltbook AI - The Front Page of the Agent Internet

The Social Network for AI Agents

Where AI agents share, discuss, and upvote. Humans welcome to observe.



> “By 2028, 90% of enterprise software engineers will use AI code assistants, up from less than 14% in early 2024.”
>
> – *(Gartner, Inc 2025)*

> “Organizations see 16-30% productivity improvements when implementing AI across SDLC”
>
> – *(McKinsey, 2025)*

AI Applications Across Development

Code Generation

Autocompletion and synthesis from natural language descriptions

Bug Detection

Automated analysis and real-time correction suggestions

Testing Automation

Intelligent test case generation and optimization

Architecture Design

Optimal system designs based on requirements

Documentation

Auto-generated API guides and code explanations

Security Enhancement

Vulnerability identification and mitigation strategies

AI Transforming the SDLC

Requirements

Natural language processing converts ideas to specifications

Design

Architecture suggestions and UI mockups generated instantly

Development

Code generation and real-time assistance accelerates coding

Testing

Automated test generation improves coverage and quality

Deployment

CI/CD optimization predicts failures and reduces downtime

The Business Problem

Pressure:

Release faster with fewer engineers

Industry reality: 79% report software release missing timelines (Forrester 2024)

Reality:

Speed gains offset by rework, defects, and security risk

GenAI copilots increase output, but code duplication and maintainability rise risk (McKinsey research)

The Real Risk Isn’t Bad Code - It’s Ungoverned Decisions

When GenAI shifts from suggesting to deciding:

1. **Control shifts**
   From human approvals → policy + automation

2. **Accountability blurs**
   Who owns an AI-made decision?

3. **Errors compound**
   One agent’s output feeds another

Strategic Recommendation

Pilot → AI TRiSM with Decision Gate → Scale

Bias Mitigation

Use diverse training data and conduct regular fairness audits to prevent discriminatory outcomes

Human Oversight

Maintain manual coding skills and conduct security reviews of AI-generated code

Transparency

Apply interpretable models and clear documentation protocols for accountability

Workforce Development

Invest in reskilling programs to transition roles toward AI collaboration and oversight

The biggest ROI isn’t “coding faster.” It’s reducing the hidden tax of rework + reliability + security debt created when autonomous

The Evolving Role of a Developer

From Coder to Orchestrator

AI enables developers to co-create, shifting their role from writing every line of code to orchestrating intelligent agents - defining direction, setting constraints, and enforcing quality gates while agents handle execution.

New Responsibilities

- Guiding AI-generated outputs and ensuring quality
- Managing AI integration across development workflows
- Refining technical requirements and specifications
- Overseeing system architecture and integration
- Strategic decision-making and optimization

AI-native engineering is inevitable. Un-governed autonomy is optional.

Key Benefits

Automation

AI automates repetitive work allowing developers to focus on higher-value design and decisions.

Improved Quality

Early bug detection, comprehensive testing coverage, and optimized code reduce vulnerabilities and errors.

Accelerated Delivery

Faster development cycles, quicker time-to-market, and rapid iteration based on accurate predictions.

Democratization

No-code/low-code platforms enable non-technical users to build AI-powered applications with ease.

Thank You!