Intelligent Software Engineering

Introduction to Artificial Intelligence

Zhilei Ren

Dalian University of Technology

September 25, 2025

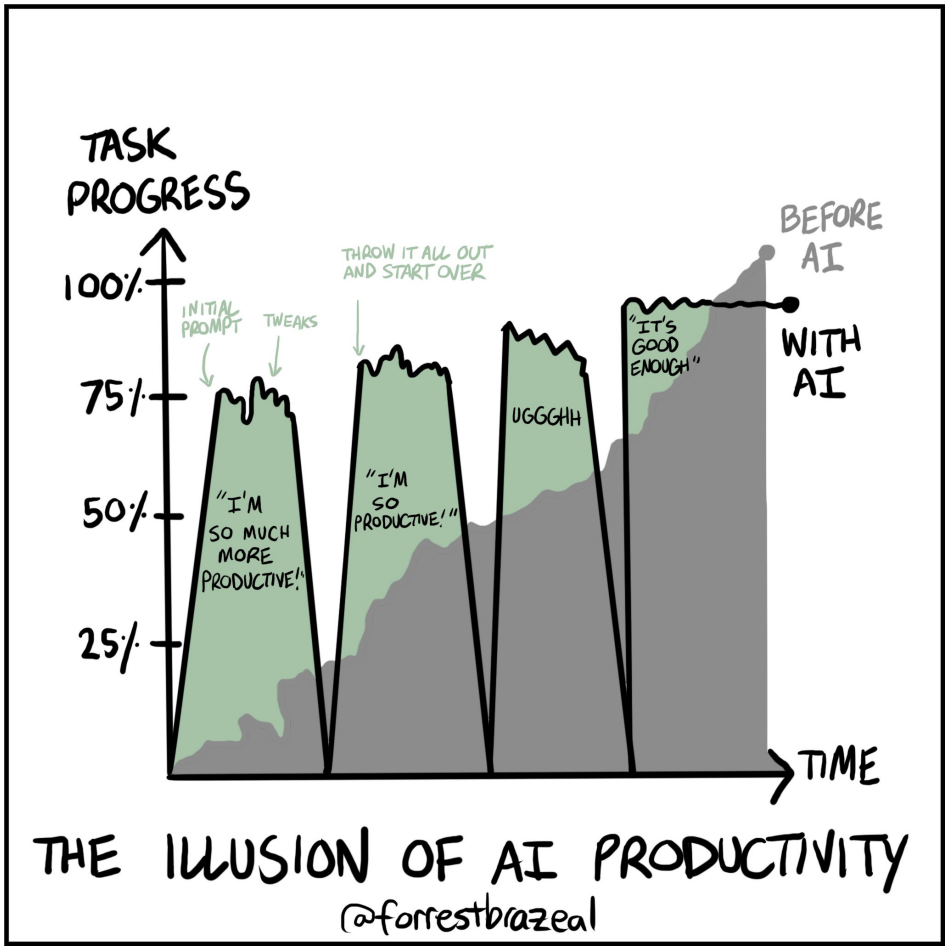
bug or feature?

Vibe Coding

Vibe coding is an artificial intelligence-assisted software development style popularized by Andrej Karpathy in February 2025. The term was listed in the Merriam-Webster Dictionary the following month as a “slang & trending” term. It describes a chatbot-based approach to creating software where the developer describes a project or task to a large language model (LLM), which generates code based on the prompt. The developer evaluates the result and asks the LLM for improvements. Unlike traditional AI-assisted coding or pair programming, the human developer avoids micromanaging the code, accepts AI-suggested completions liberally, and focuses more on iterative experimentation than code correctness or structure[1].

[1] https://en.wikipedia.org/wiki/Vibe_coding

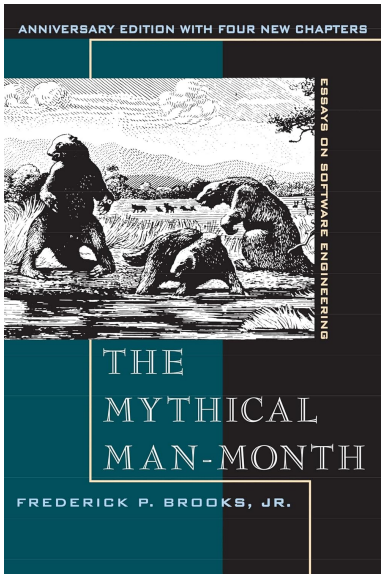
The Illusion of AI Productivity



The Mythical Man-Month

There is no single development, in either technology or management technique, which by itself promises even one order of magnitude [ten-fold] improvement within a decade in productivity, in reliability, in simplicity.

– Fred Brooks



Outline

What is PlantUML?

- **Definition:** Open-source tool for creating diagrams from plain text descriptions
- **Core Idea:** Uses simple, human-readable Domain Specific Language (DSL)
- **Foundation:** Java-based tool leveraging Graphviz for layout
- **Philosophy:** Focus on content rather than manual layout

  "PlantUML is a versatile component for quickly and directly creating diagrams."

Key Advantages

Advantage	**Description**
Version Control Friendly	Text files work with Git - enables change history, diffing, collaboration
Efficiency & Speed	Faster than manual graphical editing, especially for complex diagrams
Maintainability & Consistency	Easy updates and consistent styling with themes
Automation & Integration	Integrates with documentation pipelines, build systems, CI/CD

UML Diagrams Supported

- Sequence Diagram
- Use Case Diagram
- Class Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- State Diagram
- Object Diagram

Visual Example:

[Diagram placeholder]

Beyond UML Diagrams

- Architectural Diagrams (C4 model)
- Entity Relationship Diagrams (ERD)
- Wireframes / UI Mockups (salt library)
- Gantt charts for project management
- Mind Maps for brainstorming
- JSON/YAML visualization
- Network diagrams

How PlantUML Works

1. **Write:** Create text file (.puml) with PlantUML syntax
2. **Process:** Java processor parses text, converts to Graphviz DOT language
3. **Render:** Layout engine generates final image
4. **Output:** Get image in desired format (PNG, SVG, etc.)

Text $\rightarrow$ **PlantUML** $\rightarrow$ **Graphviz** $\rightarrow$ **Diagram**

Syntax Example: Sequence Diagram

```
@startuml
actor User
participant "Web Browser" as Browser
participant Server

autonumber
User -> Browser: Enter URL
Browser -> Server: HTTP Request
Server -> Server: Process Request
Server --> Browser: Return HTML
Browser --> User: Display Page
@enduml
```

- `@startuml`/`@enduml`: Diagram boundaries
- `actor`, `participant`: Element declarations
- `->`, `-->`: Solid/dashed arrows
- `autonumber`: Automatic message numbering

Syntax Example: Use Case Diagram

```
@startuml
left to right direction
actor "Library User" as User
usecase "Borrow Book" as Borrow
usecase "Search Catalog" as Search

User --> Borrow
User --> Search
@enduml
```

- `left to right direction`: Layout control
- `actor`, `usecase`: Actor and use case definitions
- `->`: Connection arrows

Getting Started with PlantUML

Online Servers (Quick Start):

- Official server: plantuml.com
- Community site: planttext.com
- No installation required

Local Installation (Recommended):

- Prerequisites: Java JRE + Graphviz
- IDE Plugins: VSCode, IntelliJ, Eclipse
- Command Line: Use `plantuml.jar`

Summary & Key Takeaways

- **Why PlantUML?** Version control, automation, efficiency
- **Text-Based:** Simple, readable language for diagrams
- **Wide Support:** Comprehensive UML + additional diagram types
- **Easy Integration:** Fits modern development workflows
- **Quick Start:** Online editors local integration

Embrace efficient diagram creation and maintenance!

Thank You & Questions

Resources

- Official Website: plantuml.com
- Online Demo: plantuml.com/plantuml
- Documentation: plantuml.com/guide

Questions?