Blitzy: The Dawn of System 2 AI Engineering

Autonomous Software Development through Inference-Time Reasoning



Blitzy represents the first enterprise platform engineered specifically for ‘Inference-Time Scaling’—moving beyond the constraints of real-time copilots to solve immense architectural challenges.

NotebookLM

EXECUTIVE SNAPSHOT

Blitzy

FOUNDING DNA

- **Sid Pardeshi** (Ex-NVIDIA, 25+ GenAI patents, Cloud Gaming Inventor)
- **Brian Elliott** (Serial Entrepreneur, West Point Systems Engineer)

BACKING & CAPITAL

$4.4M

SEED ROUND

- Link Ventures
- Bessemer
- Flybridge
- NFX

MARKET SIGNAL

The "Why Now"

"Compress 6-month projects into 6-day deliverables."

86.8%

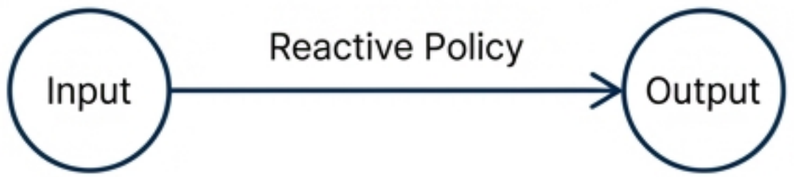
SWE-bench Verified Score

Industry Leader (Previous SOTA: ~73%)

NotebookLM

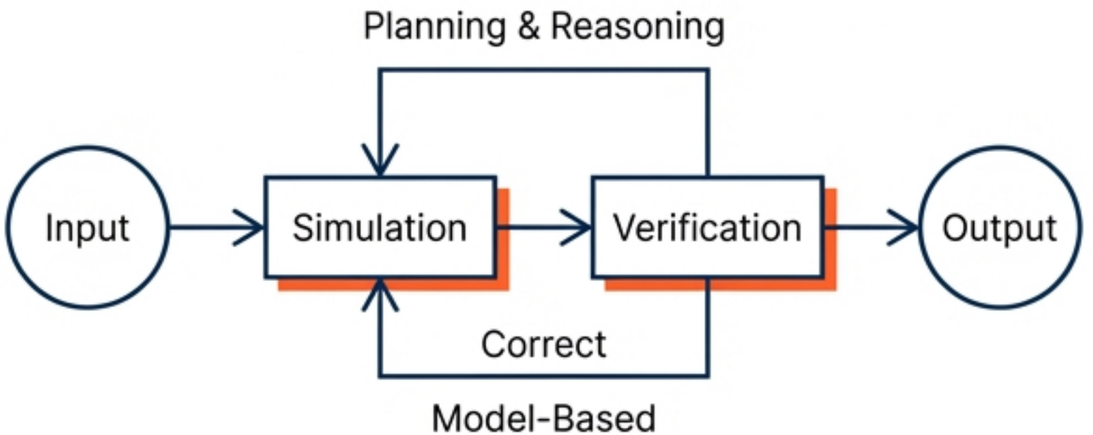
The Cognitive Limit: Why 'Fast Thinking' Fails at Scale

System 1 (Current AI)



Fast, Intuitive, Automatic.
Used by Copilots/Cursor.
Prone to hallucination.
Zero planning time.

System 2 (Blitzy Paradigm)



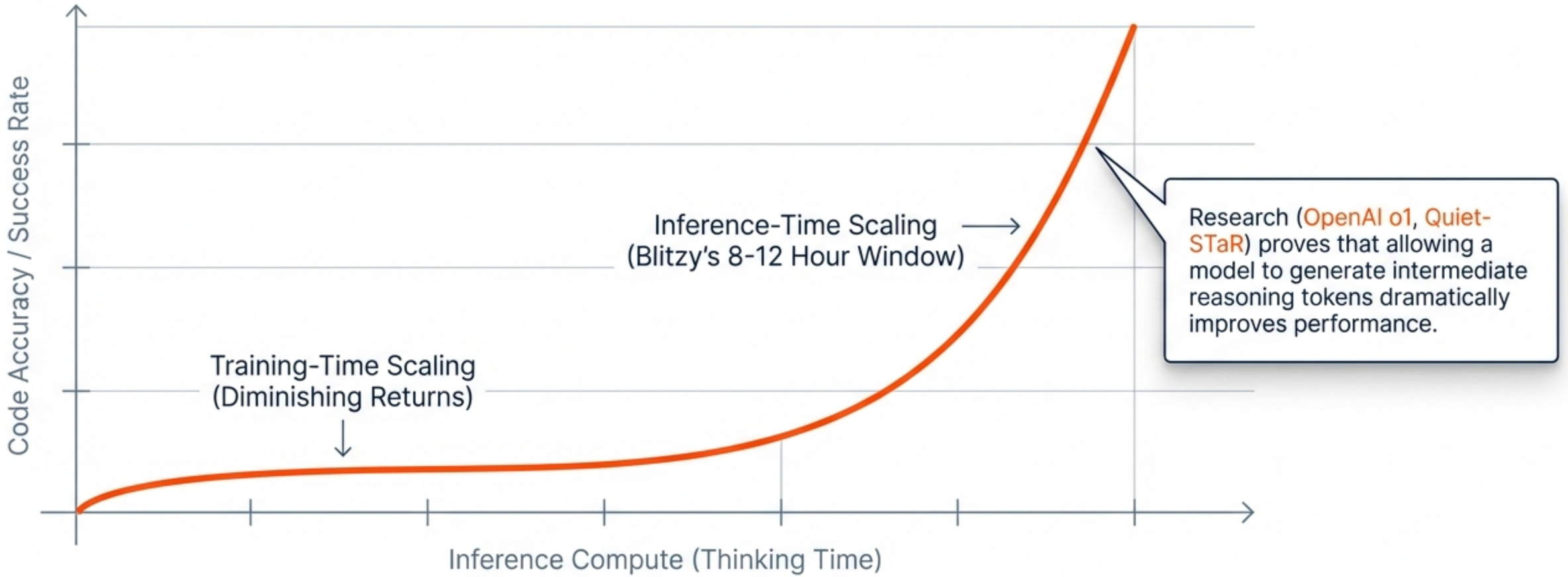
Slow, Deliberate, Conscious.
Required for complex architecture.
Uses simulation and verification before outputting code.

The Gap: Current LLMs operate on a predetermined track. They lack the luxury of time to simulate, verify, and correct.

NotebookLM

Technical Editorial • Swiss International Style

The Physics of Intelligence: Inference-Time Scaling



Blitzy's Bet: Unlike a chatbot that must reply in seconds, Blitzy takes 8-12 hours to explore solution paths, self-correct, and verify against the entire codebase.

NotebookLM

The Solution: Autonomous Batch Building

Don't guess in 2 seconds. Reason for 12 hours.



PRDs, Technical Specs,
Repo Access

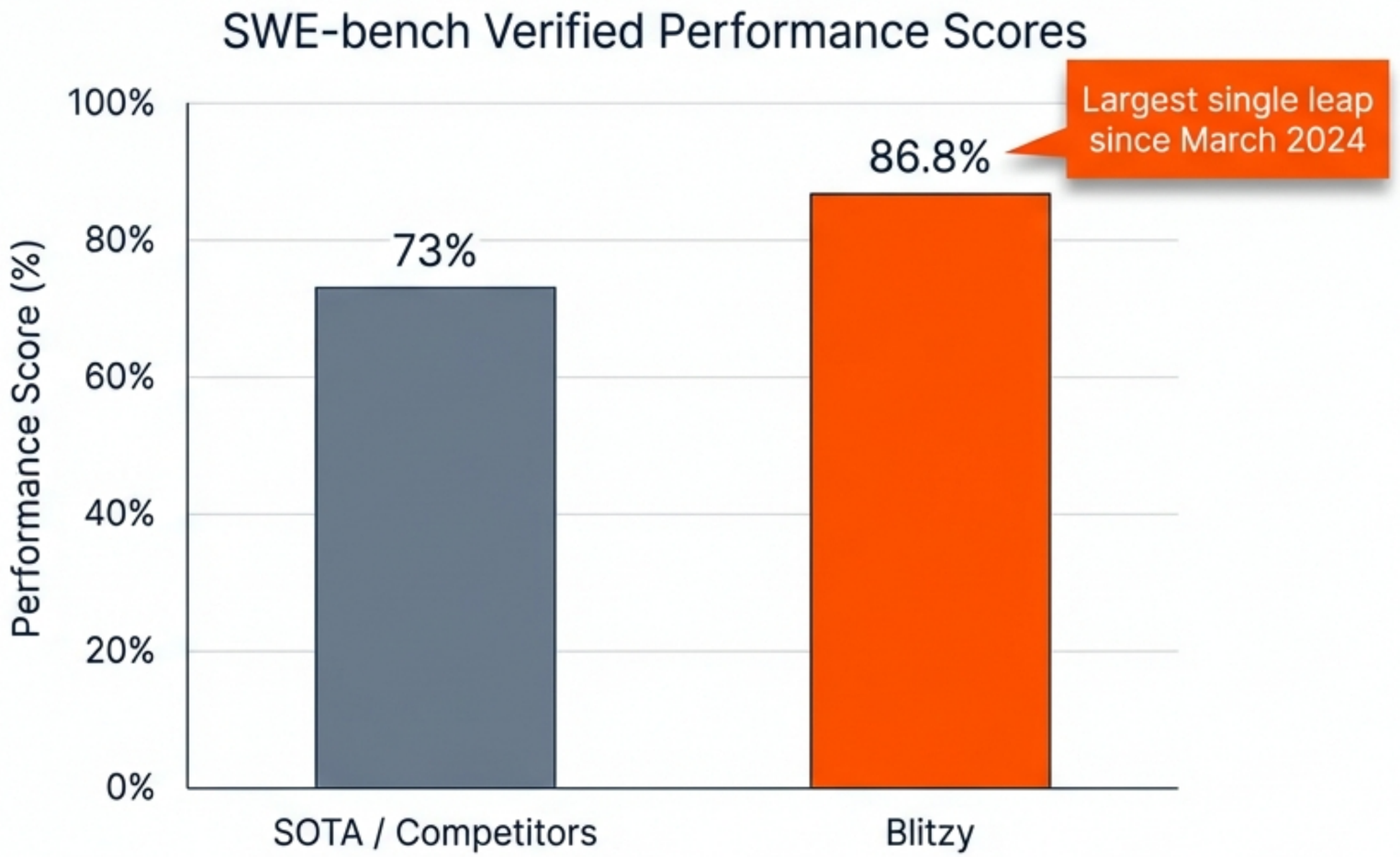
8-12 Hours. Asynchronous,
Multi-Agent Work.
"While you sleep".

Pull Request.
80% Complete.
Pre-compiled & Pre-tested.

Differentiation: Unlike inline suggestions, Blitzy creates entire features, refactors architectures, and writes documentation.

NotebookLM

Validating the Thesis: The SWE-bench Breakthrough



Context & Methodology

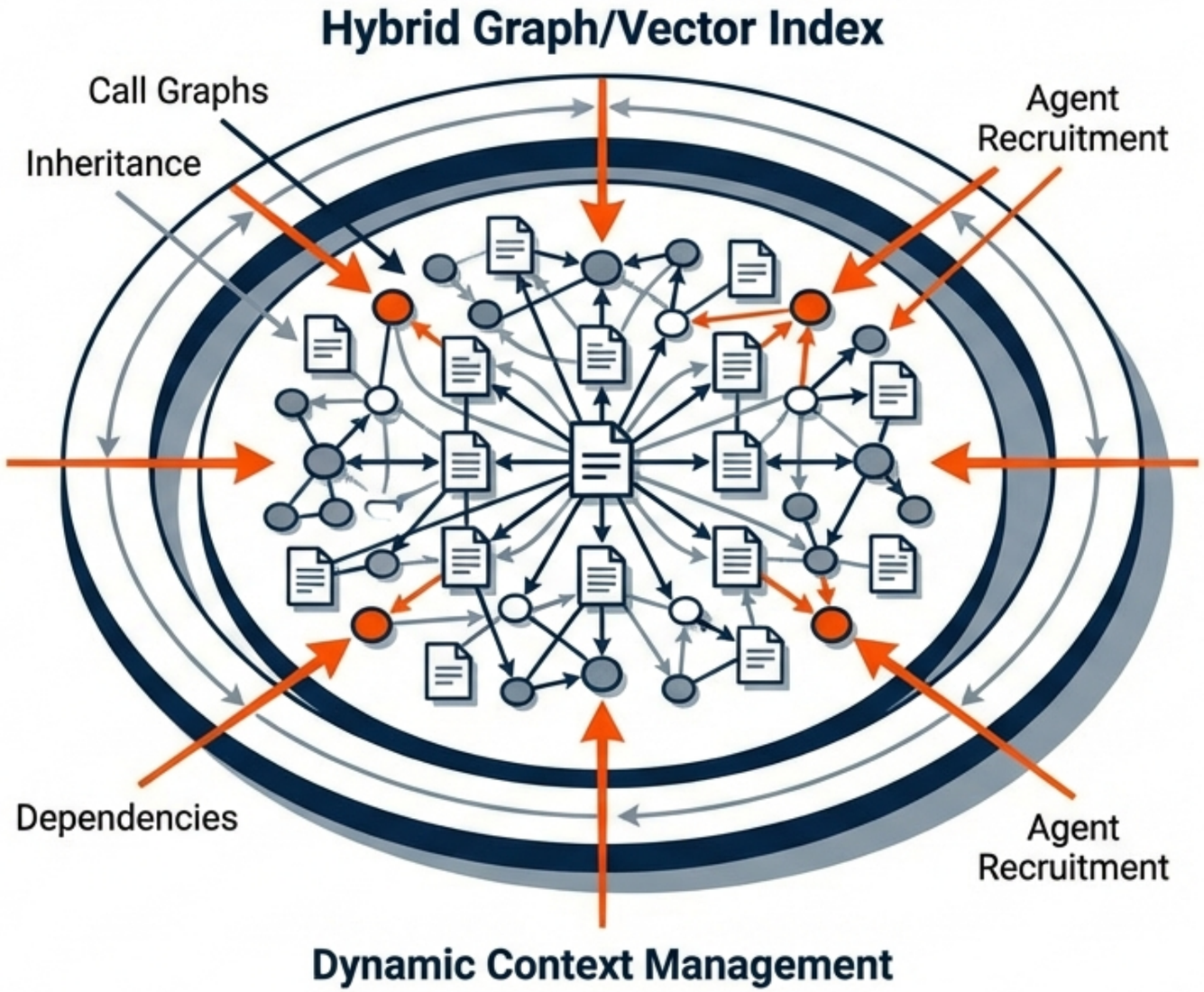
- Solved 30 of 74 historically unsolved tasks.
- Achieved via single-shot patching without ‘best-of-k’ scaffolding.
- Mirrors real production constraints (no training set leakage).

SWE-bench is the gold standard for autonomous engineering, testing an agent’s ability to navigate repositories and pass tests.

NotebookLM

Under the Hood: Infinite Code Context

The Problem:
Standard LLMs choke on large context windows, forgetting logic buried in distant files.

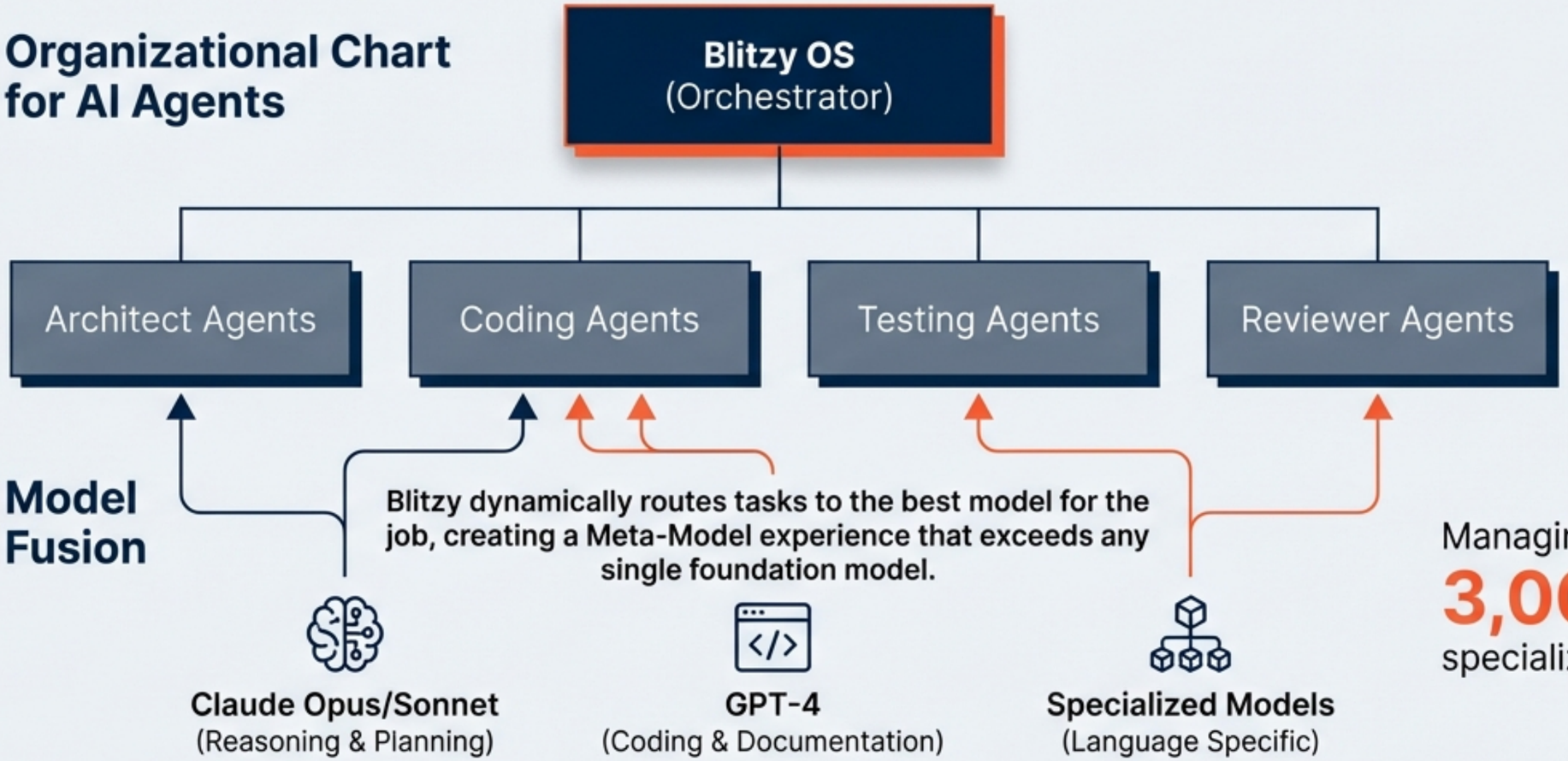


The Solution:
Virtualizes the Abstract Syntax Tree (AST).

Effectively handles 10M to 100M+ line codebases by recruiting only relevant context.

NotebookLM

The Workforce: Multi-Agent Orchestration & Model Fusion



NotebookLM

Primary Use Case: Legacy Modernization



Legacy
(COBOL)

Autonomous Translation



Modern
(Java 21)

Case Study: Credit Card Processing System

- **Timeline: 8.5 Hours Autonomous Processing**
- **Output:** 660 commits across 328 files
- **Result: 95% Completion Rate**
- **Logic:** Mathematically equivalent, infrastructure modernized.

Solving the “COBOL Cliff”: Financial systems run on dying languages.
Blitzy preserves business logic while moving to modern infrastructure.

NotebookLM

The Architect’s Janitor: Tech Debt & Security

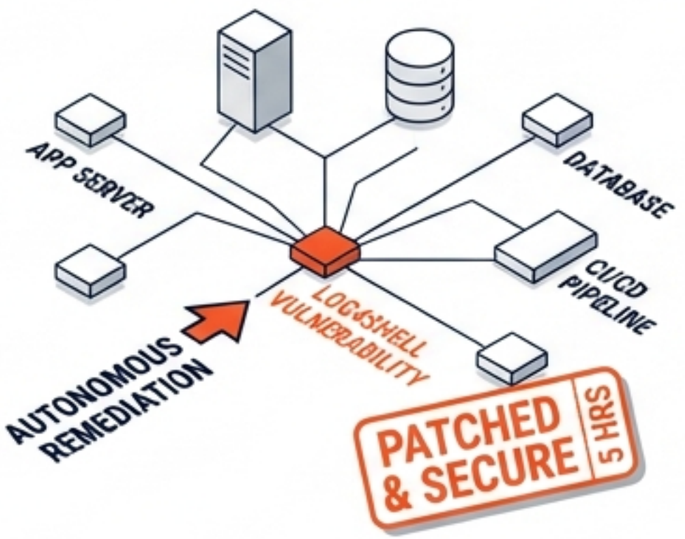
Automating the work humans hate but businesses require.

Case Study: Log4Shell Remediation

- **Severity:** 10/10 Critical
- **Human Estimate:** Weeks of manual panic
- **Blitzy Actual:** Fixed in 5 hours autonomously

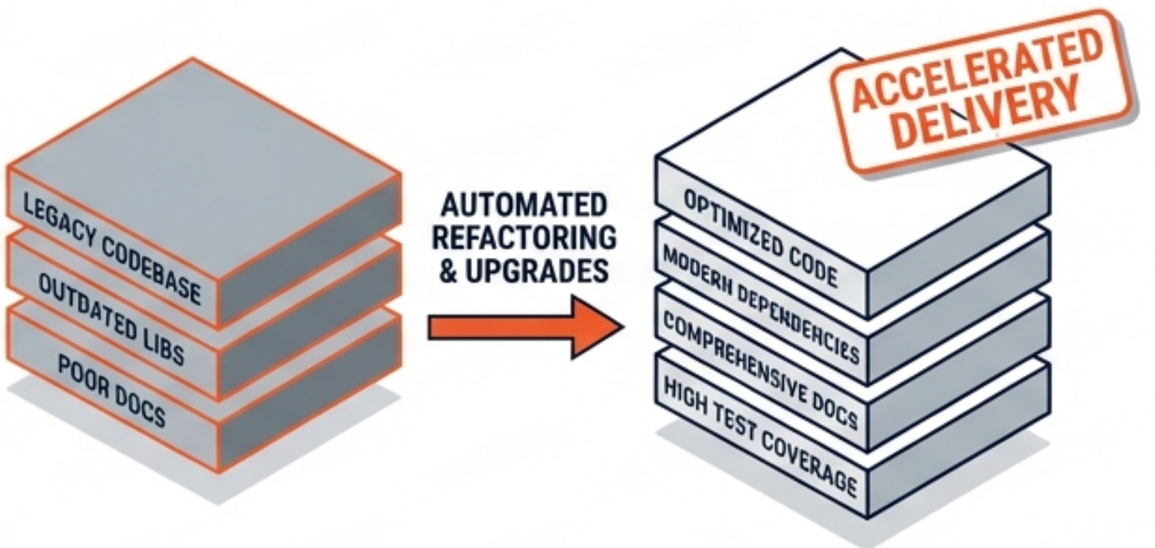
Action:

- Analyzed code, implemented patch, generated tests, documented changes.



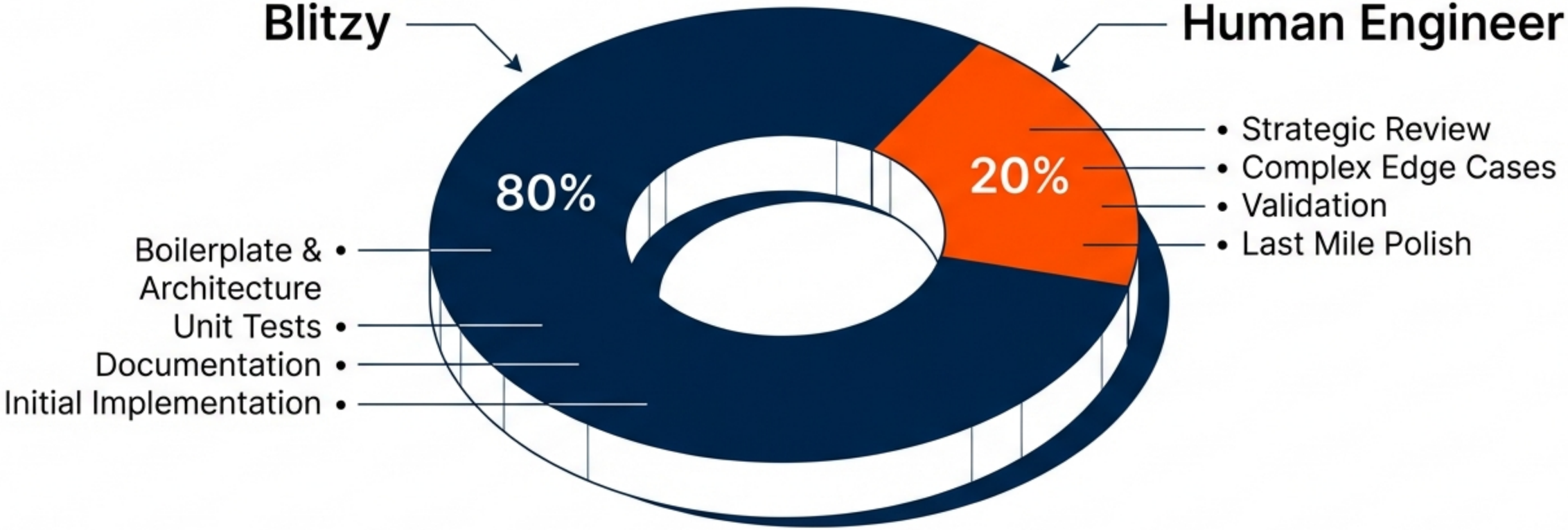
Tech Debt Reduction

- **Capabilities:** Dependency upgrades, documentation generation, test coverage expansion.
- **Impact:** Compresses 3-5 year backlogs into 12-24 months.



NotebookLM

The New Workflow: The 80/20 Rule



"The Shift: From 'Writing Code' to 'Managing Agents'. Engineers start with a substantially complete codebase rather than a blank file."

NotebookLM

Solving the UI Gap: Agent Ready Components (ARC)

The Challenge: AI typically struggles to generate consistent, brand-compliant UIs.

Figma Design Intent

ARC Library

- First component library designed for agents.
- Figma Integration: Maps design intent directly to code.
- AI Documentation: Components are documented "by AI for AI".

SYSTEM 2 ACTIVE

Production Code

```
<Button variant="primary">
Submit</Button>
```

Production-ready front-ends that respect enterprise design systems.

NotebookLM

Strategic Positioning: Complementary, Not Competitive

System 1 (Real-Time Copilots)	System 2 (Autonomous Platforms)
• Examples: GitHub Copilot, Cursor • Use Case: Inline suggestions, syntax help, small functions • Role: The Assistant • Cost: ~$10/month	• Example: Blitzy • Use Case: Legacy modernization, tech debt, new features • Role: The Batch Builder • Cost: $100k - $500k/year

Synergy: Use Blitzy for the heavy batch build, use Copilot for the 20% refinement. Validated by Galatea Associates in regulated financial environments.

NotebookLM

Enterprise Readiness & Considerations



Security

SOC 2 Type I certified. Air-gapped deployment options available for Finance/Defense.



Cost Reality

High compute cost due to inference hours. Value must be measured against engineering time saved.



Human-in-the-Loop

Not “fire and forget”. Requires senior engineers to review AI-generated architecture.



Latency

Asynchronous batch processing. Not designed for real-time iteration loops.

NotebookLM

The Future of Engineering

Relentless automation in the AI software development lifecycle.

The Trajectory



Near Term:
Infinite context expansion beyond code (Slack, Docs)

Medium Term:
Vertical-specific models (FinTech, Gov)

Long Term:
Full autonomous SDLC—ideation to deployment

Moving from the Era of the Tool User to the Era of the Agent Manager.

NotebookLM