

# Blitzzy AI: A Comprehensive Company Analysis

## Table of Contents

- Introduction: A New Architect for Enterprise Software
- The Vision: "System-2" Thinking for Autonomous Software Development
  - The Problem with "System-1" AI in Enterprise Coding
  - Blitzzy's Paradigm Shift: Inference-Time Compute & "System-2" AI
  - Core Technological Pillars
  - Validation and Performance
- Product Deep Dive: The Blitzzy Autonomous Development Platform
  - Core Capabilities & Use Cases
  - The Human-in-the-Loop Workflow (The 80/20 Model)
  - Technical & Security Specifications
  - Pricing & Engagement Model
- Market Strategy & Competitive Landscape
  - Target Market & Go-to-Market Strategy
  - Competitive Positioning: A New Category
- Company Profile: Leadership, Funding & Vision
  - Founding Team & Expertise
  - Funding and Investor Confidence
  - Future Outlook & Long-Term Vision
- Conclusion: Assessing the Blitzzy Proposition



# Introduction: A New Architect for Enterprise Software

In the rapidly evolving landscape of artificial intelligence, a new class of company is emerging—one that moves beyond assistive roles to claim a stake in autonomous creation. Blitz, a Cambridge, Massachusetts-based startup founded in 2023, stands at the forefront of this movement. It is a Generative AI company with a singular, ambitious goal: to automate the development of complex, custom enterprise software. Born from the collaboration of two Harvard masters students, [Blitz aims to transform software projects that traditionally take months or years into tasks completed in days or weeks.](#)

This report provides a deep-dive analysis of Blitz's technology, market strategy, competitive positioning, and future potential. It synthesizes publicly available data from company statements, technical benchmarks, and market comparisons to construct a comprehensive profile. The central thesis of this analysis is that Blitz should not be viewed as an incremental improvement over existing AI coding tools like co-pilots. Instead, it represents a potential paradigm shift towards what its founders call "System-2" AI—a form of slow, deliberate, and multi-step reasoning applied through an agentic architecture. This approach enables autonomous development for the large-scale, legacy systems that form the backbone of global enterprises, a challenge that has largely remained intractable for previous generations of AI tools.



## The Vision: "System-2" Thinking for Autonomous Software Development

To understand Blitz is to understand its core technological philosophy. The company's differentiation is not merely in its features, but in its fundamental approach to how AI should solve complex problems. This section analyzes the "System-2" concept that underpins Blitz's platform, explaining why it is positioned as a potential game-changer in software engineering.

### The Problem with "System-1" AI in Enterprise Coding

The first wave of popular AI coding tools, most notably GitHub Copilot, operates on a principle analogous to "System-1" thinking—fast, intuitive, and automatic. These tools excel at code completion, generating boilerplate, and suggesting solutions for localized problems. They function as highly effective pair programmers, augmenting a developer's immediate

workflow. However, their architecture presents inherent limitations when faced with the realities of enterprise software.

These limitations include:

- **Limited Context Window:** Most AI models can only process a finite amount of information (tokens) at once. This makes it impossible for them to grasp the full complexity of a multi-million-line codebase, with its intricate dependencies and historical architectural decisions.
- **Lack of Architectural Reasoning:** Co-pilots are designed for single-pass, immediate suggestions. They cannot perform the deep, architectural analysis required to modernize a legacy system, extract a microservice from a monolith, or migrate an entire application between programming languages. These tasks require sustained, multi-step reasoning, not just pattern matching.
- **Focus on Incremental Change:** The co-pilot paradigm is built around assisting with patches and small feature additions. It is not engineered to orchestrate a comprehensive redevelopment from requirements gathering through to deployment, a process that Blitzzy aims to automate. As Blitzzy notes, it is not a co-pilot; it surpasses them by [autonomously generating nearly-complete code repositories, not just snippets.](#)

## Blitzzy's Paradigm Shift: Inference-Time Compute & "System-2" AI

Blitzzy's response to these challenges is a fundamental shift in methodology. The company has built its platform around "System-2" thinking, a concept borrowed from cognitive science that describes slow, deliberate, and analytical reasoning. In the context of AI, this means moving away from the demand for instantaneous responses and instead allowing AI agents to "think" for extended periods—hours or even days. Blitzzy calls this "inference time scaling."

This approach posits that the bottleneck for solving truly complex problems is not the inherent capability of AI models, but the time they are allotted to reason. By removing this constraint, Blitzzy unlocks a new tier of problem-solving. As CTO and Co-founder Sid Pardeshi explained, this philosophy was key to their breakthrough performance on industry benchmarks:

*"The 'unsolvables' weren't actually unsolvable — they just required deeper thinking than System-1 AI could provide. By design, our platform enables AI to think for hours or days rather than seconds or minutes, unlocking solutions to problems that stumped every previous approach. This validates inference time*

*scaling as the key to exponential capability improvements."* — [Sid Pardeshi, Blitzy CTO, Yahoo Finance](#)

This paradigm of "inference time compute" is presented as the next major scaling frontier for AI, offering potentially unlimited improvement potential bounded only by computational budget, unlike the diminishing returns seen in pretraining scaling. It is this deep, sustained reasoning that allows Blitzy to tackle architectural transformation rather than just code completion.

## Core Technological Pillars

Blitzy's "System-2" vision is executed through two primary technological pillars: an agentic architecture and the ability to handle virtually infinite code context.

### Infinite Code Context

A cornerstone of Blitzy's platform is its ability to ingest, map, and understand an entire enterprise codebase, regardless of size. The company claims it can onboard repositories with "20 million lines, a hundred million lines" and maintain full context. This is achieved through a [patent-pending code ingestion framework](#) that creates a comprehensive representation of the code, analyzing relationships down to individual variables. This overcomes the fundamental context window limitation of standard Large Language Models (LLMs) and is what enables the platform to perform complex, system-wide refactoring and modernization tasks with semantic integrity.

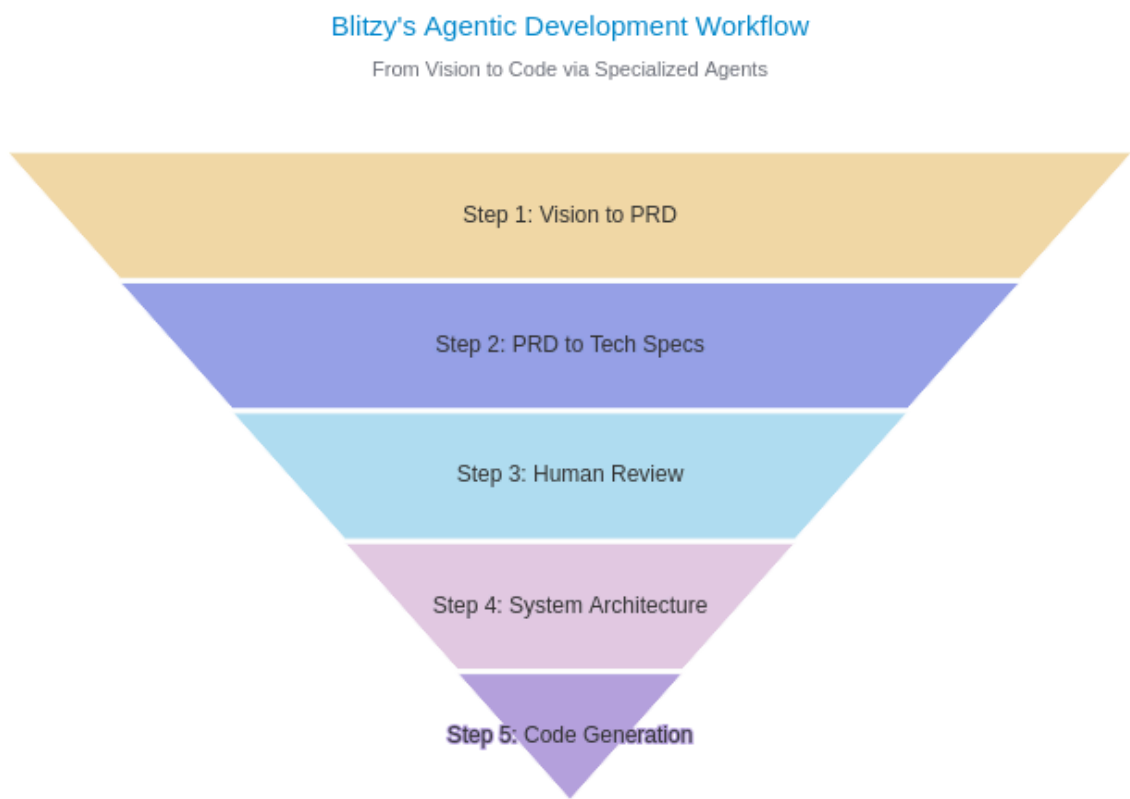
### Agentic AI Architecture

Blitzy does not rely on a single monolithic AI model. Instead, it orchestrates a cooperative of thousands of specialized AI agents that work together to execute a complex software development project. This multi-agent system breaks down the development process into a structured, sequential workflow, as detailed in an analysis by [Experimentation Machine](#):

1. Vision to Requirements (Agent 1): A specialized agent takes the initial product vision and translates it into a formal Product Requirements Document (PRD).
2. Requirements to Technical Specs (Agents 2 & 3): Subsequent agents ingest the PRD and expand it into detailed technical requirements and then into technical specifications, a step traditionally handled by senior engineers. This output is then reviewed and approved by the human client.

- 3. Architecture Design (Agent 4): Another agent translates the approved specifications into a complete system architecture, mapping out components, dependencies, and integrations.
- 4. Collaborative Code Generation (Thousands of Agents): Finally, a swarm of up to 3,000 coding agents works in parallel to build the software, each handling a specific portion of the codebase. This collaborative effort can generate hundreds of thousands, or even millions, of lines of code in a single run.

This entire process, which can take around twelve hours, dedicates the vast majority of its time (ten hours) to the initial "thinking" stages of planning and architecture, with only the final two hours spent on actual code generation. This deliberate, front-loaded reasoning is the essence of Blitzy's "System-2" approach in action.



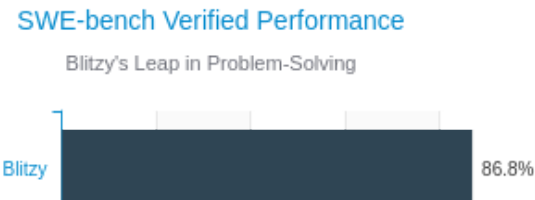
Source: Analysis based on [Experimentation Machine](#)

**Validation and Performance**

A philosophical vision requires empirical validation. Blitzy has provided this through both standardized benchmarks and real-world enterprise case studies.

**SWE-bench Breakthrough**

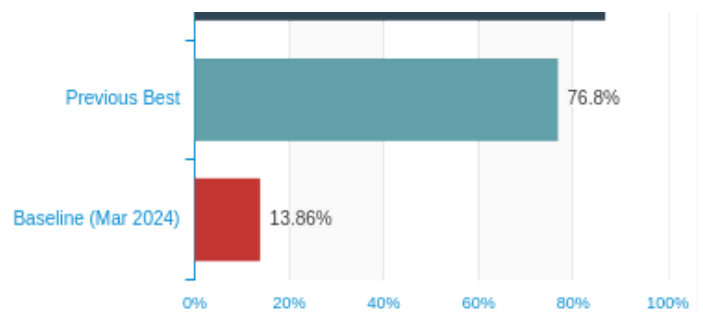
The most significant public validation of Blitzy's methodology is its performance



on SWE-bench Verified, a challenging benchmark that tests an AI's ability to resolve real-world GitHub issues in Python repositories. Blitzzy achieved a score of 86.8%, a result that represented a 10 percentage point leap over the previous best score of 76.8%. [This was](#)

[described as the largest single advance](#)

since the benchmark's inception. While some research has highlighted potential issues like solution leakage in static benchmarks, Blitzzy's performance was framed not just as a high score, but as a powerful validation of its "System-2," inference-time compute approach. It demonstrated that by allowing AI to "think" longer, it could solve problems that had previously been considered "unsolvable" by faster, "System-1" methods.



Data Source: [Yahoo Finance](#)

### ***Enterprise-Scale Proof***

Beyond benchmarks, which often focus on isolated problems, Blitzzy's true value proposition lies in its application to large-scale enterprise challenges. The company has cited several powerful examples of its platform in action:

- **Rapid Code Migration:** A recent case study highlighted the refactoring of a 30,000-line codebase from MATLAB to Python in just 48 hours—a task that would typically take a development team months to complete.
- **Legacy System Modernization:** The platform was used to modernize a 4-million-line legacy Java application, a task that involved over 72 hours of distributed reasoning time for major architectural decisions.
- **Monolith Decomposition:** Blitzzy's agents performed over 24 hours of architectural analysis to identify optimal boundaries for extracting services from a 500,000-line monolith.

These examples, which are far beyond the scope of current benchmarks, serve as the ultimate proof of Blitzzy's enterprise-grade capabilities and the power of its sustained AI reasoning model.

### ***Key Takeaways: The "System-2" Vision***

- Blitzzy positions itself as a paradigm shift from "System-1" (fast, intuitive) AI co-pilots to "System-2" (slow, deliberate) autonomous agents.

- The core innovation is "inference time scaling," allowing AI agents to reason for hours or days to solve complex architectural problems.
- This is enabled by two pillars: "infinite code context" to understand entire codebases and a multi-agent system that mimics a full development team.
- The approach has been validated by a breakthrough 86.8% score on SWE-bench and successful modernization of multi-million-line enterprise applications.



## Product Deep Dive: The Blitzy Autonomous Development Platform

While the "System-2" philosophy provides the 'why', the Blitzy platform itself is the 'what'. This section provides a detailed breakdown of the platform's capabilities, the customer workflow, its technical specifications, and the engagement model through which enterprises can adopt this technology.

### Core Capabilities & Use Cases

Blitzy is not a general-purpose coding tool; it is an "autonomous software development platform" engineered for specific, high-value enterprise tasks that are often the most painful and resource-intensive. Its primary use cases include:

- **Legacy System Modernization:** This is arguably Blitzy's flagship use case. The platform is designed to tackle aging but critical systems, such as mainframe applications or software built on outdated Java or COBOL stacks. It can refactor these systems to modern technologies while preserving the core business logic, a task fraught with risk when done manually.
- **Large-Scale Code Refactoring:** Blitzy can execute complete cross-language migrations (e.g., MATLAB to Python, Java to Go) or framework upgrades (e.g., AngularJS to React). It claims to maintain mathematical precision and semantic equivalence through extended verification cycles.
- **New Feature Development on Existing Products:** The platform can add new, complex features to existing products by first ingesting and understanding the current codebase, ensuring that new functionality aligns with existing patterns and standards.

- **Autonomous Documentation:** A significant byproduct of its process is the generation of comprehensive documentation. Blitzzy can create system-level technical design documents and code-level documentation, addressing a common pain point in enterprise software maintenance and regulatory compliance.

## The Human-in-the-Loop Workflow (The 80/20 Model)

Blitzzy is explicit that its goal is to "supercharge"; human engineers, not replace them. The platform operates on an 80/20 model, where AI autonomously handles approximately 80% of the development work, leaving the final 20% of "last-mile" development and creative refinement to the human team. The workflow is designed to keep the human expert in a strategic, oversight role.

The customer's role in the process is clearly defined:

1. **Provide the 'Why' and 'What':** The process begins with the customer providing the initial requirements. Blitzzy recommends a structured `WHY/WHAT/HOW` prompt format to ensure clarity of vision, core requirements, and technical constraints.
2. **Approve the Technical Specification:** After Blitzzy's agents generate the detailed technical specification, the customer's engineering team reviews, edits, and approves this document. This is a critical human-in-the-loop checkpoint that ensures the AI's plan aligns perfectly with the desired outcome before code generation begins.
3. **Complete the Final 20%:** Once Blitzzy delivers the generated codebase (which is pre-compiled and validated), the human team takes over to handle the final integration, perform final QA/QC, and add the nuanced, creative touches that require human ingenuity.

As CEO Brian Elliott states, "The final touches—the creativity, the innovation—still rely on the brilliance of human engineers. Blitzzy is designed to handle the heavy lifting." This collaborative model aims to elevate developers from tedious implementation to high-value strategic work.

## Technical & Security Specifications

For its enterprise target market, technical robustness and security are non-negotiable. Blitzzy's platform is built with these requirements at its core.



- **Language Support:** The platform is language-agnostic and [works with all programming languages](#), making it versatile for the heterogeneous technology stacks found in large organizations.
- **Security & Compliance:** Blitzzy emphasizes its enterprise-grade security posture. The platform is [built on SOC 2 Type II certified infrastructure](#) and also holds an ISO 27001 certification. For maximum security, it offers air-gapped enterprise deployments to ensure complete data isolation. Crucially, Blitzzy guarantees that it does not train its AI models on customer code.
- **Input & Control:** Users maintain control over the AI's operation. In addition to the structured prompting and spec approval process, developers can create a `.blitzzyignore` file in their repository, similar to `.gitignore`, to explicitly exclude certain files or folders from being processed by the AI agents.
- **Validation:** A key differentiator from simple code generators is Blitzzy's validation process. The platform compiles and performs runtime validation on the code it generates to ensure it works correctly before delivery, providing what it calls "enterprise-grade validation."

## Pricing & Engagement Model

Blitzzy's pricing is structured to align with the enterprise procurement process, allowing for validation before a large-scale commitment. It uses a two-phase model: Evaluation followed by Deployment.

### *Phase 1: Evaluation*

This phase is designed to let enterprises validate the ROI at their preferred scale. There are three options:

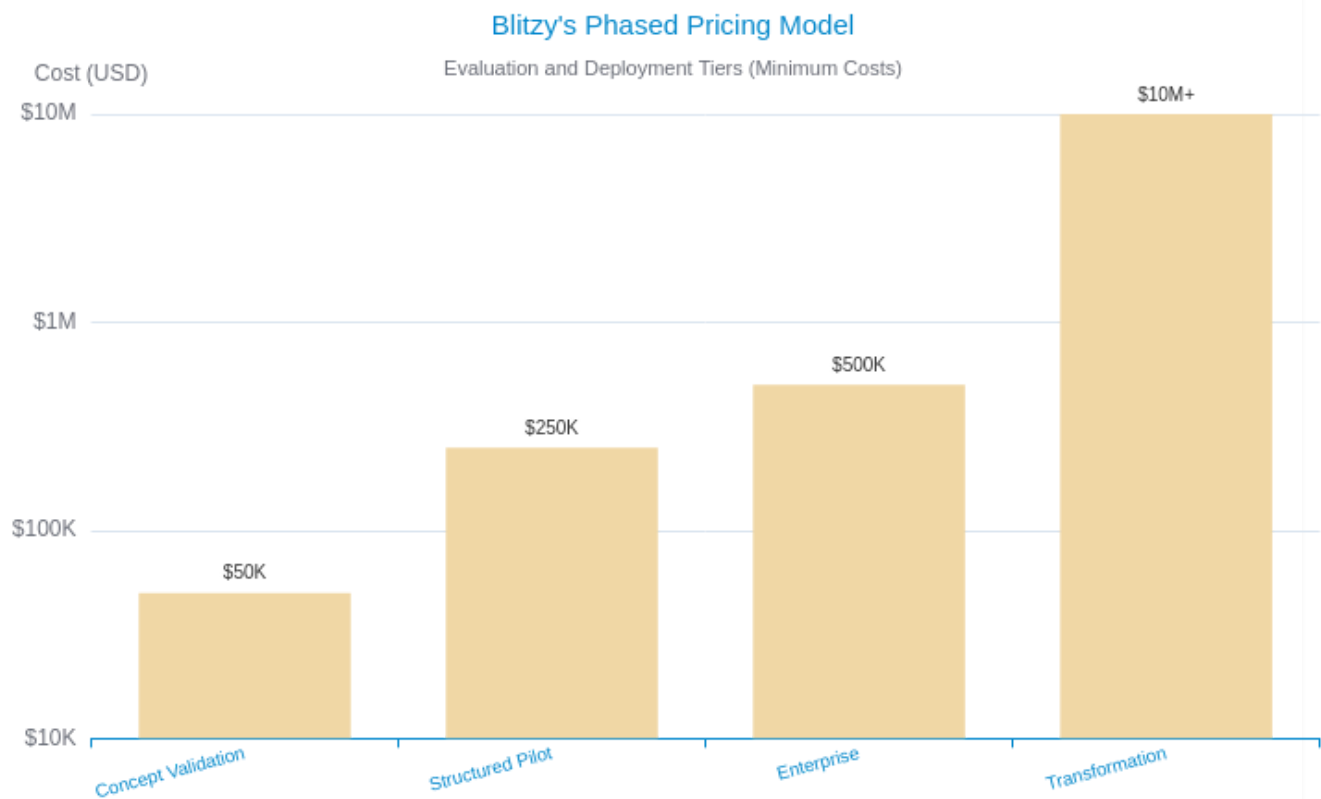
- **Explore (\$0):** A risk-free initial assessment to understand the platform's capabilities.
- **Concept Validation (\$50,000):** A 2-month term where Blitzzy delivers a guided Proof of Concept (POC) to demonstrate value on a specific problem.
- **Structured Pilot (\$250,000):** A 6-month term that involves a full deployment of Blitzzy in the client's environment, with onboarding of up to 5 million lines of code and generation of 1.25 million lines to prove production readiness.

### *Phase 2: Deployment*

Following a successful evaluation, organizations can move to a long-term deployment path:

- Enterprise (\$500,000+ per year): A 36-month term that handles onboarding of up to 50 million lines of code, with generation capacity of 2.5-15 million lines. This tier includes dedicated infrastructure and SAML-SSO.
- Transformation (\$10,000,000+ per year): A 48-month term for organizations seeking complete digital transformation. It provides infinite code context, custom deployment options, and embedded Forward Deployed Engineering teams from Blitzy.

Underpinning these tiers is a usage-based model of [\\$0.20 per line of code generated](#), which directly aligns the cost with the value delivered.



Source: [Blitzy Official Website](#)

## Market Strategy & Competitive Landscape

A groundbreaking technology is only as impactful as its ability to reach the right market. Blitzy's go-to-market strategy and competitive positioning are as deliberate as its technical architecture, focusing on a specific customer segment and carving out a new category for itself.

### Target Market & Go-to-Market Strategy

#### *Ideal Customer Profile*

Blitzzy is not for everyone. Its ideal customer is a large enterprise grappling with massive, complex, and often aging codebases. These are organizations where technical debt is a significant drag on innovation and where modernization projects are deemed too risky, too expensive, or too slow to undertake with traditional methods. Key target sectors include:

- Financial Services: Banks, insurance companies, and trading firms that rely on decades-old mainframe and legacy systems.
- Professional Services: Large consulting firms that can leverage Blitzzy to accelerate client projects.
- Technology: Established tech companies looking to modernize their own monolithic software products.

### ***Strategic Partnerships: The Galatea Example***

A cornerstone of Blitzzy's strategy is forming deep partnerships with domain experts. The most prominent example is its [strategic partnership with Galatea Associates](#), a technology consulting firm with 25 years of experience in the financial services industry. This partnership combines Blitzzy's autonomous development platform with Galatea's deep domain knowledge of securities financing, payment systems, and wealth management platforms.

This collaboration is designed to overcome the trust barrier in a risk-averse industry. To underscore its confidence, the partnership offers a powerful guarantee: if Blitzzy fails to demonstrate a 5x faster development velocity compared to the client's traditional approaches during a POC, the engagement does not proceed. This "5x or no fee" model is a potent market entry strategy, mitigating risk for potential clients and forcing a focus on measurable outcomes.

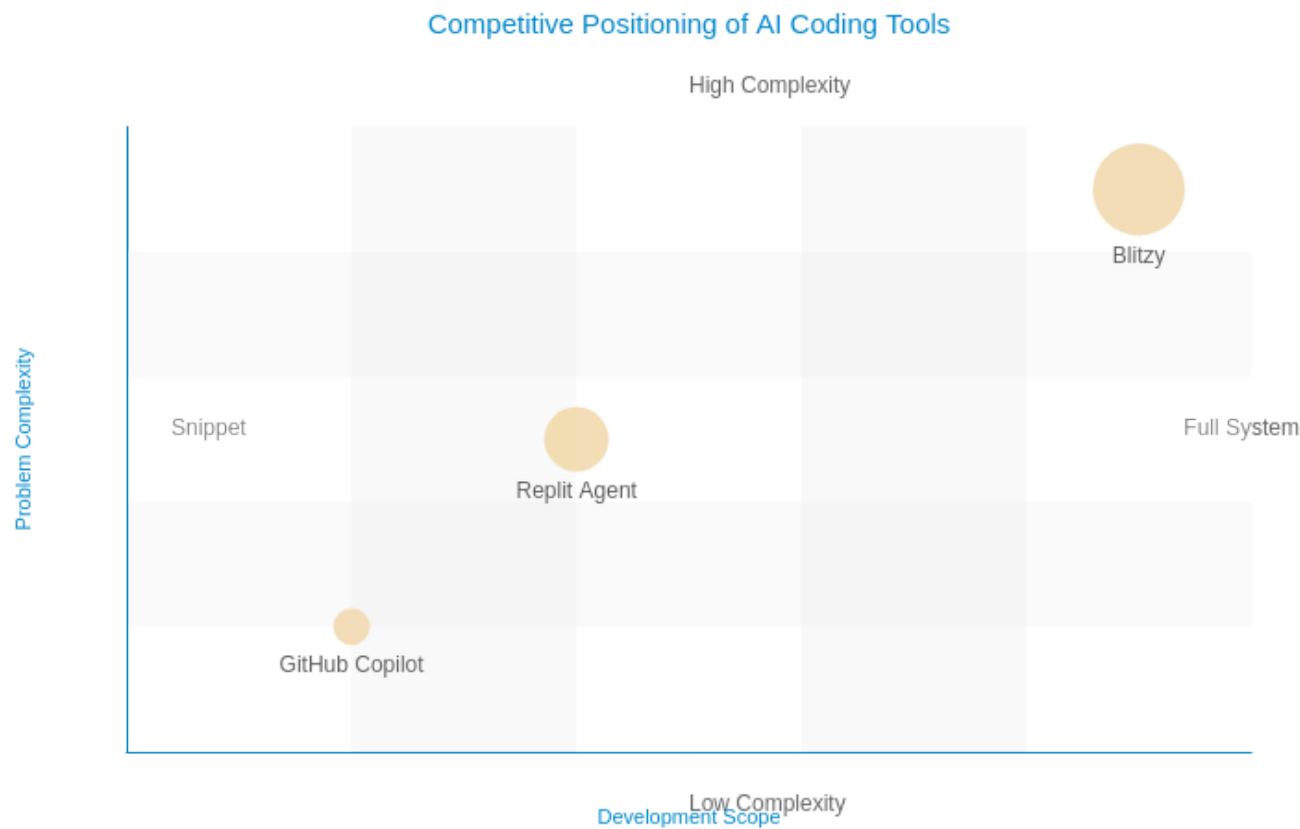
### ***Market Traction***

The strategy appears to be working. According to CEO Brian Elliott, the company is witnessing a "seismic shift" in its market engagement. Initially, Blitzzy had to pursue clients, but now, [Fortune 100 companies are actively seeking them out](#). This shift from outbound sales to inbound interest suggests that the pain points of legacy modernization are becoming acute at the highest levels of enterprise leadership, and Blitzzy's message of exponential productivity is resonating.

### ***Competitive Positioning: A New Category***

Blitzzy carefully positions itself not as a competitor to existing tools, but as the creator of a new category. It differentiates itself by focusing on a different, more complex layer of the

software development stack.



Source: Analyst interpretation of provided market data.

**Blitzzy vs. AI Co-pilots (e.g., GitHub Copilot)**

This is the most crucial distinction. Blitzzy is not a co-pilot. While co-pilots assist with line-by-line or function-by-function coding, Blitzzy operates at the architectural and system level. A useful analogy is that a co-pilot is an AI assistant suggesting the next word or sentence for an author, whereas Blitzzy is an autonomous team tasked with writing the entire book, from outline to final draft. The focus is on end-to-end repository generation and architectural transformation, not snippet completion.

**Blitzzy vs. Cloud IDEs with AI (e.g., Replit)**

Cloud-based Integrated Development Environments (IDEs) like Replit are also leveraging agentic AI. However, their focus and use case are different. Based on detailed comparisons, [Replit is optimized for rapid prototyping, full-stack web application development, and real-time collaborative coding](#) in a browser-based environment. It excels at turning an idea into a functional app quickly. Blitzzy, in contrast, is built for heavy-duty, industrial-scale enterprise work. Its strength lies in tackling the complexity of a 4-million-line Java monolith, a task that is fundamentally unsuited for an IDE-based agent designed for web development.

**Blitzzy vs. Other Agentic Platforms (e.g., Codeium, Lovable)**

Blitzzy is not the only company building agentic platforms. Competitors like Replit Agent, Codeium, and Lovable are also gaining traction. However, Blitzzy's key claimed differentiators are its "infinite code context" and its deep "System-2" reasoning capability. It positions itself as a solution for a higher class of complexity. While other platforms might build a new application from a prompt, Blitzzy's core marketing emphasizes its unique ability to understand and transform massive, existing, and highly complex legacy systems—a domain where few, if any, competitors make similar claims.

## **Company Profile: Leadership, Funding & Vision**

The credibility of a startup's ambitious vision is often anchored in the strength of its leadership, the confidence of its investors, and the clarity of its long-term goals. Blitzzy's profile suggests a strong foundation in all three areas.

### **Founding Team & Expertise**

Blitzzy was co-founded by a team whose backgrounds blend deep technical expertise with strategic business acumen and disciplined execution.

- **Brian Elliott (CEO):** Elliott's profile is one of leadership and strategy. A graduate of Harvard Business School and a serial entrepreneur, he brings business-building experience to the company. His background as a US Army Ranger and a West Point graduate suggests a foundation in disciplined execution and strategic planning, qualities essential for navigating the complex enterprise market.
- **Siddhant ";Sid" Pardeshi (CTO):** Pardeshi provides the deep technical credibility. A former software architect at NVIDIA, he holds over 27 Generative AI patents and was recognized with a Master Inventor award. His dual MS/MBA degree from Harvard complements his technical depth with a strong understanding of business application. This powerful combination of technical invention and business strategy is at the heart of Blitzzy's identity.

### **Funding and Investor Confidence**

In 2024, Blitzzy secured significant early-stage funding, signaling strong investor confidence in its vision and technology. The company raised a \$4.4 million Seed round. While the amount is typical for a seed stage, the quality of the investors is notable. The round included backing from established venture capital firms such as Bessemer Venture Partners and Flybridge Capital Partners. The participation of such well-regarded VCs provides crucial third-party validation of Blitzzy's team, technology, and market opportunity.

## Future Outlook & Long-Term Vision

Blitzzy's ambitions extend far beyond being a tool for code modernization. Its leadership team articulates a vision that could reshape enterprise technology and knowledge work itself.

### The Roadmap Acceleration

The immediate vision, as articulated by CEO Brian Elliott, is one of radical productivity acceleration. He challenges CIOs and CTOs to rethink their planning horizons:

*"If you can do 5x [the work], are you actually gonna do 2025, 2026, 2027 roadmap all this year? Because that's what we can unlock."* — [Brian Elliott, Blitzzy CEO](#)

This is the core of Blitzzy's value proposition: compressing multi-year digital transformation roadmaps into a matter of months, thereby creating an immense competitive advantage for its customers.

### Beyond Coding

The company sees its "System-2" AI paradigm as a general-purpose reasoning engine applicable to any complex, multi-step domain. The founders have indicated that the same approach of sustained, deep reasoning could be applied to fields like [medical diagnosis, financial analysis, legal research, and engineering design](#). This suggests a long-term ambition to evolve from a software development company into a broad platform for enterprise-scale AI reasoning.

### The Ultimate Goal

The ultimate vision is a world where the constraints of generic, off-the-shelf software are a thing of the past. Blitzzy envisions a future where [all B2B software is custom-built to the exact specifications of the customer](#), enabled by autonomous AI development platforms like its own. This represents a fundamental democratization of software creation, moving from a world of mass-produced tools to one of bespoke, perfectly-fitted solutions.

## Conclusion: Assessing the Blitzzy Proposition

Blitzzy AI has emerged not merely as another tool in the crowded AI landscape, but as a proponent of a new philosophy for software engineering. This analysis has shown that the

company's proposition is built on a cohesive and compelling foundation, from its "System-2" technological vision to its targeted market strategy and experienced leadership.

The core value proposition is clear and potent: Blitzy promises not just incremental efficiency but exponential productivity. It offers enterprises a path to tackle the intractable challenges of legacy modernization that have accumulated for decades, turning a source of immense technical debt into an opportunity for competitive advantage. By automating the 80% of development that is heavy lifting, it aims to free human engineers to focus on the 20% that involves true creativity and innovation.

The company has successfully backed its bold claims with tangible evidence—a record-breaking benchmark score, credible enterprise case studies, and a risk-mitigated engagement model that forces it to prove its value. Its strategic positioning, which carves out a new category of autonomous development for complex systems, effectively sidesteps direct competition with the more commoditized co-pilot and cloud IDE markets.

Ultimately, Blitzy stands as a key player in the emerging "Agentic AI Revolution." It is testing a critical hypothesis: that the next great leap in AI's utility will come not from faster models, but from models that are given the time and architecture to think deeply. The central question for the industry is whether this model of sustained, multi-agent AI reasoning will become the new standard for enterprise software engineering. If it does, Blitzy is exceptionally well-positioned to be one of the architects of that future.

## Reference

- [1] About Blitzy: AI-Driven Software Innovation  
[https://blitzy.com/our\\_story](https://blitzy.com/our_story)
- [2] Blitzy - Crunchbase Company Profile & Funding  
<https://www.crunchbase.com/organization/blitzy>
- [3] Blitzy: AI-Powered Autonomous Software Development Platform  
<https://blitzy.com/>
- [4] Blitzy Blows Past SWE-bench Verified, Demonstrating Next ...  
<https://finance.yahoo.com/news/blitzy-blows-past-swe-bench-123000094.html>
- [5] Blitzy Platform Released  
<https://www.devopsdigest.com/blitzy-platform-released>
- [6] Blitzy CEO Outlines Enterprise AI Strategy Focused on ...  
<https://www.tipranks.com/news/private-companies/blitzy-ceo-outlines-enterprise-ai-strategy-focused-on-inference-time-compute-and-modernization>
- [7] How Blitzy Works: AI Autonomous Code Platform

[https://blitzzy.com/how\\_it\\_works](https://blitzzy.com/how_it_works)

[8] Blitzzy: AI-Powered Autonomous Software Development Platform

<https://blitzzy.com/>

[9] Professional Dota

[10] The Real Impact of Agentic AI [Case Study]

<https://experimentationmachine.com/p/the-real-impact-of-agentic-ai-case-study>

[11] Blitzzy Partners with Galatea Associates to Deliver AI-Native ...

<https://www.prnewswire.com/news-releases/blitzzy-partners-with-galatea-associates-to-deliver-ai-native-development-for-financial-services-302482765.html>

[12] Replit vs GitHub Copilot: Full Comparison Guide

<https://replit.com/discover/replit-vs-github-copilot>

[13] Blitzzy Raises \$4.4 Million in Funding

<https://www.thesaasnews.com/news/blitzzy-raises-4-4-million-in-funding>

[14] Blitzzy

<https://www.linkedin.com/company/blitzyai>

[15] Blitzzy - 2025 Company Profile, Team, Funding & Competitors

[https://tracxn.com/d/companies/blitzzy/\\_\\_\\_oiylGuevdaHOcMdhR2-ILsxT8cWpz8IKDw2xv4cLALI](https://tracxn.com/d/companies/blitzzy/___oiylGuevdaHOcMdhR2-ILsxT8cWpz8IKDw2xv4cLALI)