國立臺灣大學管理學院企業管理碩士專班

碩士論文

Global MBA

College of Management

National Taiwan University

Master's Thesis

CodeDocs 商業計劃：透過 AI 與程式碼穿越技術, 打造智慧化代碼庫映射方案

Codedocs Business Plan: Pioneering Intelligent Codebase Mapping Through AI and Code Traversal

賴孟欣

Albert Lai

指導教授：堯里昂 博士

Advisor: Leon van Jaarsveldt, Ph.D.

中華民國 114 年 7 月

July 2025

doi:10.6342/NTU202502429

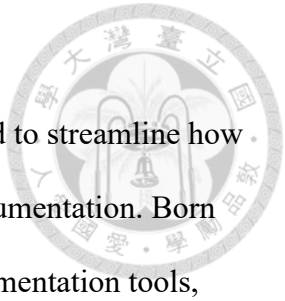
Acknowledgement

After meeting as colleagues in 2018, Victor Vong and I quickly developed a close friendship grounded in mutual respect and a shared passion for building meaningful products. While working at Zoom, Victor reached out with an idea that would eventually become CodeDocs — a collaborative documentation platform for engineers. The opportunity to co-found CodeDocs was born. I'm grateful to Victor not only for his vision and partnership, but also for the trust and camaraderie that made this journey both ambitious and deeply fulfilling.

I would also like to extend my sincere thanks to Professor Leon Van Jaarsveldt. Though our paths didn't cross in the classroom, his reputation for clarity and rigor preceded him. I was fortunate to have his mentorship during this thesis process, and I am deeply appreciative of the insight, generosity, and steady guidance he brought as my advisor.

CodeDocs would not exist without the dedication of Mike Bart and Ilynn Chao — our Chief Technology Officer and founding designer, respectively. From building the platform's technical foundation to crafting a thoughtful user experience, their contributions were instrumental in helping us bring CodeDocs to market. Their tireless work, creativity, and commitment not only shaped the product, but also made this journey even more meaningful. I'm proud to be building alongside this team and excited for what lies ahead.

doi:10.6342/NTU202502429



Executive Summary

CodeDocs is a collaborative documentation platform designed to streamline how software engineering teams create, share, and maintain technical documentation. Born out of firsthand experience with the inefficiencies of traditional documentation tools, CodeDocs aims to bridge the gap between engineering workflows and product documentation by integrating real-time collaboration, structured templates, and intelligent version control into a seamless, developer-first experience.

This thesis explores the opportunity landscape for modern developer tools, with a particular focus on how documentation pain points impact team productivity, onboarding, and product velocity. Through user interviews, early prototyping, and market validation, we identified key user behaviors and unmet needs among software engineers, product managers, and technical writers. The findings informed iterative product development and guided the design of a minimum viable product that has since been deployed in pilot environments.

The thesis also outlines the formation of the founding team, the technology stack used to build the core platform, and the initial go-to-market considerations. CodeDocs demonstrates the potential of verticalized tools that are deeply aligned with developer workflows. This project serves as both a case study in early-stage venture building and a reflection on the intersection of product development, team dynamics, and technical execution.

Keywords: Startup, technical documentation, developer tools, product design, Software as a Service, collaboration, CodeDocs, workflow optimization, early-stage venture

doi:10.6342/NTU202502429

摘要

CodeDocs 是摘要一個協作式文件平摘要台，旨在簡化軟體工程團隊建立、分享與維護技術文件的方式。此平台誕生於對傳統文件工具效率低落的親身體驗，致力於透過即時協作、結構化範本以及智慧版控功能，打造一個無縫接軌且以開發者為中心的使用體驗，彌補工程流程與產品文件之間的斷層。

本論文探討現代開發工具的市場機會，特別著重於文件相關痛點如何影響團隊生產力、成員培訓與產品開發速度。透過用戶訪談、早期原型設計與市場驗證，我們發現了軟體工程師、產品經理與技術寫手之間的關鍵使用行為與未被滿足的需求。這些洞察推動了產品的迭代開發，並指導我們設計出最小可行產品，目前已部署於多個試點環境中。

本論文亦闡述創辦團隊的組成、建構核心平台所採用的技術堆疊，以及初步市場進入策略。CodeDocs 展現出與開發者工作流程深度整合的垂直化工具的潛力。本專案不僅是一個早期創業實踐的個案研究，也反映了產品開發、團隊協作與技術執行三者之間的交會與互動。

關鍵字: 新創公司, 技術文件, 開發者工具, 產品設計, 軟體即服務, 協同合作, 工作流程優化, 早期創業項目

doi:10.6342/NTU202502429

Table of Contents

doi:10.6342/NTU202502429

doi:10.6342/NTU202502429

1. Business Description

1.1 Founders

CodeDocs was co-founded by Victor Vong and Albert Lai, two colleagues-turned-close-friends who met while working in Silicon Valley. Victor, who served as a Staff Software Engineer and Tech Lead at Zoom, originally conceived the idea after experiencing firsthand the frustrations of modern code documentation practices. He approached Albert, an experienced venture partner and serial entrepreneur, to co-find a solution that would reshape how software teams manage internal knowledge and technical documentation.

1.2 Core Business Concept

CodeDocs was founded on the belief that the way engineers document, navigate, and share knowledge across codebases is fundamentally broken. At its core, CodeDocs seeks to transform code documentation from a static, manual afterthought into a living, dynamic layer built into the code itself. The vision is to eliminate context-switching for engineers by allowing them to document, query, and explore code execution paths—all from within the environments they already use. CodeDocs is designed for a new generation of teams who value efficiency, collaboration, and intelligence in their development tools.

1.3 Current Status

The company has completed its early product development phase and is currently testing its platform with pre-committed beta users across 34 companies. The platform's minimum viable product has been implemented in controlled environments to gather data, refine user experience, and validate the effectiveness of its core technology—the traversal engine. The team continues to iterate on the product based on feedback from

doi:10.6342/NTU202502429

engineers, Chief Technology Officers, and technical teams within its early adopter network.

1.4 Main Product Offerings

The heart of CodeDocs is its code traversal engine, a proprietary tool that dynamically maps how code executes across functions, files, and folders, no matter how complex or fragmented the system. This engine integrates with a developer's code editor (such as VSCode) and enables instant navigation across a codebase. Combined with artificial intelligence powered explanations, it provides developers with an intuitive, explorable guide to any system. Additional features include inline documentation using Markdown, team-based Question & Answering system tied to specific lines of code, a sandbox lab for testing multi-repository documentation output, and smart compliance alerts to ensure key documentation Key Performance Indicators are met before code deployment. These offerings together create a cohesive, developer-first platform that transforms the way codebases are learned and maintained.

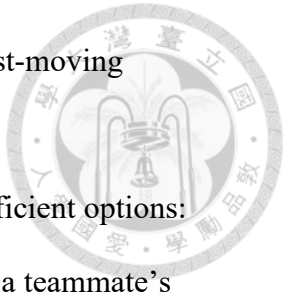
1.5 Branding

The brand identity of CodeDocs is built around clarity, collaboration, and trust. Unlike traditional tools that force engineers to write documentation on disconnected platforms like Notion or Confluence, CodeDocs integrates directly into their workflow. The platform positions itself as a bottom-up solution—built by engineers for engineers—designed to grow alongside teams and codebases. The brand messaging emphasizes its artificial intelligence explainability, self-hosting readiness for enterprise adoption, and commitment to reducing onboarding friction and documentation decay.

1.6 Problem Statement

Software companies—big and small—face costly inefficiencies due to fragmented, outdated processes for code documentation. Documentation is only

doi:10.6342/NTU202502429



accurate as long as the underlying code remains unchanged, but in fast-moving environments, code evolves constantly.

Without an integrated solution, engineers are left with two inefficient options: manually dig through repositories with hundreds of files, or interrupt a teammate's workflow to ask, "Can you explain this section?" Both approaches are time-consuming and costly. While documentation is essential to software development because it saves time when done correctly (Rachel, 2019), the real challenge lies in maintaining it. As soon as code changes, documentation risks becoming obsolete. When developers finally need that documentation, they often scramble to sync it—after the context has already been lost.

The issue becomes more acute during onboarding. New engineers often require three to nine months to become fully productive (Dror, 2020). In practice, onboarding follows a frustrating loop: teams hastily prepare documentation, new hires read it and ask questions, but the documentation is already out of sync—and the answers are forgotten. This recurring cycle leads to knowledge gaps and delays across teams.

At the organizational level, the cost of engineering churn is especially painful. Tech companies face some of the highest turnover rates of any industry (Staffing Industry Analysts, 2023). For instance, the median tenure at Google is just 1.1 years (Johnson, 2021). When senior engineers leave, they often take with them undocumented institutional knowledge that is difficult—if not impossible—to replace. Onboarding new hires takes months, and even then, most engineers operate at just 40% to 70% of their full productivity. The loss of tribal knowledge can severely affect product quality, team efficiency, and velocity.

doi:10.6342/NTU202502429

CodeDocs was created to address these problems head-on: accelerating onboarding, reducing documentation decay, and turning codebases into living, navigable knowledge systems.

2 Business Objective

2.1 Description of Products & Services

Modern software teams face an inherent tension between rapid iteration and sustainable documentation. Developers move quickly, codebases evolve daily, and the burden of maintaining documentation often becomes an afterthought. Yet, without consistent documentation, knowledge decays, onboarding slows, and team velocity suffers. CodeDocs aims to resolve this tension by embedding documentation directly into the engineering workflow — eliminating friction and turning documentation from a chore into a strategic advantage.

The traditional practice of switching between an Integrated Development Environment and external platforms like Notion or Confluence fragments the developer experience. This context switching discourages engineers from updating documentation and increases the likelihood of inconsistencies. CodeDocs eliminates that divide by enabling documentation to be generated dynamically from the codebase itself. In short, Markdown makes it easier to format text for web pages because its tags are simpler than HyperText Markup Language, and they convert to HyperText Markup Language automatically. This means you don't have to know HyperText Markup Language to write something for a web page because Markdown translates your tags into HyperText Markup Language for you (Woodgate, 2022). As the code evolves, so does the documentation — automatically and in real time.

A core technical insight behind CodeDocs is that code does not execute linearly. Software systems are modular, distributed, and cross-functional by nature. Functions are

doi:10.6342/NTU202502429

often written in one location but called from another — sometimes across teams, folders, or even services (Johnson, 2021). This complexity introduces a major limitation in current artificial intelligence models: they have a high error rate when trying to locate the true relationships between functions and predict actual execution logic. Most artificial intelligence systems read code files in isolation, without a reliable understanding of how those files interact at runtime. As a result, they frequently misinterpret dependencies, skip critical paths, or provide incomplete explanations — especially in large or dynamic codebases.

To solve this, CodeDocs introduces its proprietary code traversal engine — a system-aware tool that visualizes how code is executed across a project. It stitches together function calls, dependencies, and execution flows into an explorable interface, offering engineers a real-time map of software behavior. This foundational context not only helps humans navigate complex systems, but also provides the artificial intelligence with a clear, accurate execution blueprint. By anchoring artificial intelligence interpretation in verified logic flows, CodeDocs dramatically reduces error rates and transforms explanations into reliable, interactive guides — with natural-language summaries for any function, dependency, or execution path.

One of the highest-friction pain points in engineering organizations is onboarding. Guidebooks are often written manually — and just as often, they become obsolete shortly after being created. To address this, CodeDocs introduces guidebook markers, which allow engineers to tag and annotate relevant parts of a codebase with onboarding notes. These markers are dynamically aggregated into a living guidebook that evolves alongside the codebase. Unlike static onboarding wikis, the CodeDocs guidebook reflects the current state of the system and can even identify and remove outdated notes via artificial intelligence. If new developers have questions, they can highlight the

doi:10.6342/NTU202502429

relevant portion of code, leave a query, and route it directly to the responsible team. All queries and responses are stored for future reference, gradually building a repository of contextual knowledge.

Despite clear benefits, many engineers still avoid writing documentation altogether. A study by Wolf (2016) found that time pressure, lack of incentives, and unclear documentation standards remain persistent obstacles. CodeDocs addresses these behavioral barriers by making documentation a native part of the workflow — and by offering artificial intelligence powered enforcement tools. For example, teams can configure the platform to block code pushes if documentation fails to meet defined thresholds or Key Performance Indicators. This ensures that documentation quality becomes a shared responsibility, enforced not by process overhead, but through seamless automation.

The objective of CodeDocs is to reduce knowledge loss, shorten onboarding times, and create long-term memory for engineering teams — all while minimizing developer resistance and operational friction. In doing so, it transforms documentation from an afterthought into a strategic asset.

2.2 Market Research and Organization Management

The developer tools market is undergoing a rapid transformation, driven by the global surge in demand for engineering talent and the increasing complexity of software development. As engineering teams become the backbone of digital transformation, the tools that support them are no longer seen as cost centers but as strategic assets. Developer efficiency has become a top priority for companies competing in fast-moving markets, creating a fertile landscape for innovation in tooling.

Dawn Capital has referred to developers as the “new rock stars of Software as a Service,” underscoring the cultural and economic shift toward building products for

doi:10.6342/NTU202502429

developers as the primary users (Bankiya & Pozzato, 2021). With more than 26 million software developers active worldwide — a number expected to nearly double within the next decade (Geerligs, 2021) — the market demand for tools that enhance productivity, reduce friction, and accelerate code deployment has never been stronger.

As workloads intensify, organizations are increasingly investing in platforms that can drive efficiency without adding overhead. Even modest improvements in code clarity and onboarding speed can lead to major downstream savings in engineering time. Analysts have estimated that if every company were to invest just 1% of an engineer's salary into productivity tools, the total addressable market for developer tooling in Europe alone would exceed $8 billion (Bankiya & Pozzato, 2021).

CodeDocs is well-positioned to capitalize on this rising demand for intelligent, artificial intelligence augmented developer tools. As developer growth outpaces the available supply of experienced engineers, tools that reduce onboarding time, mitigate knowledge loss, and make codebases more explorable will become essential. The emergence of a business-to-developer market has created new funding momentum, with top-tier investors such as Y Combinator, Andreessen Horowitz, and Sequoia aggressively backing companies in this space.

The global Software Development Tools market size is projected to reach $9.9 billion by 2027, from $3.8 billion in 2020, at a Compound Annual Growth Rate of 14.5% during 2021-2027 (Industry Research, 2022). The market signals are clear: teams will increasingly rely on developer-first platforms to scale engineering output without proportionally increasing headcount. CodeDocs' differentiated approach — combining execution path traversal, inline documentation, and artificial intelligence powered explainability — uniquely positions it to serve the next generation of developer teams.

doi:10.6342/NTU202502429

Year	Number of Software Developers
2018	23.9 million
2019	26,4 million
2021	26,9 million
2023	27,7 million
2024	28.7 million
2030	45 million

Figure 1. Growth Rate of Software Engineers

Source. Geerligs, R. (2021, September 23). *How many developers are in US and in the world [updated]*. Daxx Software Development Teams. Retrieved May 26, 2022, from https://www.daxx.com/blog/development-trends/number-software-developers-world

2.3 Political, Economic, Social, Technological, Legal, and Environmental Analysis

This section evaluates the macro-environmental factors that may impact CodeDocs' ability to scale and operate across global markets. The Political, Economic, Social, Technological, Legal, and Environmental framework — which includes political, economic, social, technological, legal, and environmental factors — is especially relevant for developer Software as a Service platforms operating in highly regulated and rapidly evolving technology sectors.

2.3.1 Political Factors

CodeDocs operates in an increasingly complex regulatory landscape. Governments around the world are enacting stricter data privacy laws such as the General Data Protection Regulation in Europe and the California Consumer Privacy Act

doi:10.6342/NTU202502429

in the United States. These regulations have direct implications on how developer tools manage source code, metadata, and user-generated documentation (European Commission, 2022).

The General Data Protection Regulation is a sweeping European Union law that governs how organizations collect, process, and store personal data of individuals in the European Union. The California Consumer Privacy Act grants similar rights to California residents, including access to, deletion of, and control over their personal data. For CodeDocs, compliance with these laws means building robust systems for data anonymization, user consent, opt-out mechanisms, and the ability to purge data upon request. Given that developer platforms can process sensitive metadata and documentation, failure to meet these standards could lead to severe legal penalties and reputational damage, particularly when serving enterprise clients in healthcare, finance, or government sectors.

At the same time, the rise of open-source adoption in public institutions is creating new market entry points for platforms like CodeDocs. Governments are increasingly investing in open-source technology to reduce vendor lock-in and increase transparency, particularly for digital infrastructure projects (McBride, 2021). As a tool that can be deployed in both cloud and self-hosted environments, CodeDocs aligns well with this shift.

Geopolitical dynamics also play a role. Ongoing tensions between the U.S. and China have led to heightened scrutiny of software supply chains and third-party code integrations, particularly among globally distributed engineering teams (Lee, 2021). By providing an artificial intelligence powered code traversal engine that enhances visibility into code execution paths and flags suspicious patterns, CodeDocs can offer a

doi:10.6342/NTU202502429

compliance-friendly solution that mitigates security concerns and supports code transparency initiatives.

2.3.2 Economic Factors

Economic conditions in the tech sector are undergoing significant realignment. Developer salaries have reached all-time highs in many regions, contributing to a global scarcity of experienced engineering talent (O'Brien, 2022). For instance, senior software engineers in the United States now command average salaries exceeding $165,000, with compensation packages often climbing higher in competitive markets like San Francisco and New York (O'Brien, 2022). As companies face budget constraints and hiring slowdowns, they are increasingly turning to automation and workflow tools to improve output without expanding headcount. This makes solutions like CodeDocs — which reduce onboarding time, minimize knowledge loss, and boost productivity — highly attractive to engineering organizations looking to operate more efficiently.

Additionally, large-scale layoffs in the tech sector have pushed companies to reevaluate their operational models. Over 260,000 tech workers were laid off globally in 2023, more than double the total layoffs from the prior year (TechCrunch, 2023). Many firms are now shifting away from rapid expansion and prioritizing tools that enhance team efficiency, reduce churn, and ensure code quality without increasing management overhead. Despite market volatility, venture capital investment in developer tools has remained resilient.

doi:10.6342/NTU202502429

Table 1 Estimated market opportunity for CodeDocs (2025–2027)

Segment	2025 Estimate (USD)	2026 Estimate (USD)	2027 Estimate (USD)
Global developer tools market	$6.41B	$7.46B	$8.68B
Enterprise developer teams (20% of market)	$1.28B	$1.49B	$1.73B
Documentation tools sub-segment	$448.7M	$522.2M	$607.6M
CodeDocs targetable market (10% of sub-segment)	$44.87M	$52.22M	$60.76M
CodeDocs realistic penetration (2%)	$0.897M	$1.044M	$1.215M

Note: Market estimates are based on Mordor Intelligence (2025) data and internal assumptions. The assessment focuses solely on the developer tools market, which includes integrated development environments, productivity tools, and documentation platforms, and excludes AI products offered by CodeDocs. A conservative market capture of 2% of the addressable market is assumed by 2027.

2.3.3 Social Factors

Developer culture places a premium on autonomy, speed, and seamless tooling. Tools that add friction to a developer's workflow are often rejected, regardless of their potential utility. According to the 2023 Stack Overflow Developer Survey, over 65% of developers say they prioritize tools that "just work" and integrate directly into their existing environments — most notably, the Integrated Development Environment. CodeDocs aligns with this principle by offering native integrations into popular editors like VSCode, allowing developers to access, edit, and contribute to documentation without leaving their coding environment.

doi:10.6342/NTU202502429

This streamlined integration has measurable internal consequences. Research from McKinsey (2020) found that developers spend up to 40% of their time searching for information or understanding existing code — time that could otherwise be spent on high-leverage work. By embedding artificial intelligence powered traversal and documentation directly into their workflow, CodeDocs helps reduce this search burden, saving an estimated 8–12 hours per engineer per week. This increased efficiency not only accelerates delivery timelines but also boosts intrinsic motivation by letting developers focus on creative problem-solving rather than repetitive documentation tasks.

Furthermore, this kind of timesaving reduces cognitive load and fosters a greater sense of mastery — two key drivers of motivation and engagement as outlined in Self-Determination Theory. Developers are more likely to experiment, contribute innovative solutions, and feel psychologically safe when tools support rather than obstruct their momentum.

The shift to remote and hybrid work has only intensified these needs. Documentation is no longer a luxury — it is a necessity for distributed teams operating across time zones and departments. With remote work now normalized across global tech firms, tools that centralize tribal knowledge and provide historical context asynchronously are in greater demand than ever before (McKinsey, 2020). CodeDocs' support for inline annotations, team Q&A threads, and real-time updates helps maintain cross-functional clarity and strengthens team alignment — key factors in preserving motivation and engagement among remote developers.

2.3.4 Technological Factors

Technological advancements are reshaping the expectations developers have for their tools. The rise of artificial intelligence assisted coding platforms — such as

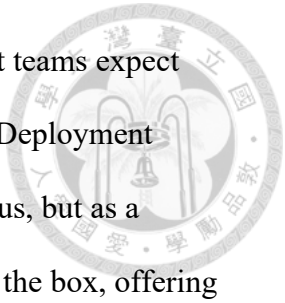
doi:10.6342/NTU202502429

GitHub Copilot, TabNine, and Amazon CodeWhisperer — has elevated the baseline for productivity-enhancing features in the development environment (Microsoft Developer Trends, 2022). Engineers increasingly expect their tools not only to support development, but to actively assist in it. CodeDocs meets this standard with its artificial intelligence powered traversal engine and explainability layer, which deliver real-time summaries of function behavior and code execution paths.

The widespread adoption of Large Language Models has further accelerated this shift. Models like Generative Pre-trained Transformer-4 and Claude have demonstrated that artificial intelligence can not only autocomplete code, but also generate boilerplate, refactor legacy systems, and explain logic in natural language. This has raised the bar for developer tools — platforms are now expected to integrate intelligent code assistance directly into the development workflow. CodeDocs stands out by narrowing the gap between raw artificial intelligence output and usable, context-specific insight. Rather than treating Large Language Models as a monolithic black box, CodeDocs contextualizes their capabilities using its traversal engine to constrain outputs to relevant code paths, significantly improving the signal-to-noise ratio.

Meanwhile, the growth of microservices, containerized infrastructure, and multi-repository architectures has increased codebase fragmentation. This has made it more difficult for developers to gain a holistic understanding of how systems behave. According to the 2023 McKinsey DevOps Report, organizations cite “difficulty in navigating distributed systems” as a top barrier to developer efficiency. CodeDocs directly addresses this issue by enabling cross-repository code visualization, function tracing, and version-based snapshots — all stitched together into an intuitive, explorable interface.

doi:10.6342/NTU202502429



Integration standards have also evolved. Modern development teams expect seamless compatibility with Git, Continuous Integration/Continuous Deployment pipelines, and Integrated Development Environments — not as a bonus, but as a requirement. CodeDocs is designed to meet these expectations out of the box, offering native support for Git workflows, Integrated Development Environment plugins, and extensible Application Programming Interfaces. The platform's architecture is built with composability in mind, allowing teams to integrate documentation intelligence into their existing toolchains with minimal friction.

2.3.5 Environmental Factors

As a cloud-native Software as a Service product, CodeDocs inherently benefits from a relatively small environmental footprint compared to traditional on-premises software solutions. Operating primarily in the cloud reduces the need for physical infrastructure, energy-intensive data centers, and localized Information Technology maintenance, contributing to lower carbon emissions and resource consumption. However, in today's enterprise landscape, sustainability considerations are no longer a "nice to have" — they are increasingly a prerequisite.

Enterprise buyers, particularly large corporations and publicly traded firms are under mounting pressure from stakeholders, regulators, and Environmental, Social, and Governance rating agencies to ensure that their vendors align with their broader sustainability and governance commitments. As such, procurement teams now regularly assess vendors on Environmental, Social, and Governance metrics, from carbon neutrality and energy usage transparency to labor practices, data privacy, and corporate governance frameworks. "In 2022, the rate of reporting among the G250 remains at 96 percent, the same as 2020" (KPMG, 2022), highlighting how Environmental, Social,

doi:10.6342/NTU202502429

and Governance reporting has become a near-universal expectation among the world's largest companies.

For CodeDocs, this means that incorporating Environmental, Social, and Governance into its core operations and messaging is not only a responsible business practice but also a strategic advantage in enterprise sales cycles. Initiatives such as adopting carbon offset programs, partnering with green cloud providers (e.g., Amazon Web Services' sustainability initiatives or Google Cloud's carbon-aware computing), and publishing an annual Environmental, Social, and Governance statement can meaningfully enhance credibility with Environmental, Social, and Governance conscious buyers. Furthermore, strong governance practices — such as ethical artificial intelligence usage, System and Organization Controls 2 compliance, and transparent data handling policies — signal long-term operational integrity and risk management.

By proactively addressing Environmental, Social, and Governance, CodeDocs positions itself not only as a technically excellent solution but also as a partner that helps enterprises meet their own sustainability goals, which can often be the deciding factor in vendor selection processes.

doi:10.6342/NTU202502429

Table 2 Political, Economic, Social, Technological, Legal, and Environmental Analysis Summary for CodeDocs

Factor	Impact Summary
Political	Tighter global data privacy laws (e.g., General Data Protection Regulation, California Consumer Privacy Act) rising scrutiny over international software integration and increasing adoption of open source in public sectors.
Economic	Developer salaries rising, post-layoff shift toward productivity tools, and continued Venture Capital interest in developer tools despite market downturns.
Social	Developer culture favors low-friction, Integrated Development Environment-native tools. Remote work has heightened the need for accessible, collaborative documentation.
Technological	Artificial intelligence driven coding tools have set new workflow standards. Code complexity from microservices increases demand for code mapping tools like CodeDocs.
Legal	System and Organization Controls 2 compliance is required for enterprise. artificial intelligence generated code introduces new legal concerns around IP, licensing, and liability.
Environmental	CodeDocs' cloud-based model has a low footprint, but Environmental, Social, and Governance minded clients may favor providers with carbon neutrality or reporting transparency.

Note: Political, Economic, Social, Technological, Legal, and Environmental analysis is based on qualitative assessment of macro-environmental trends and publicly available sources such as McKinsey, World Intellectual Property Organization, and Microsoft.

2.4 Porter's Five Forces

doi:10.6342/NTU202502429

Porter's Five Forces is a strategic framework used to assess the competitiveness and attractiveness of an industry. For a startup like CodeDocs, this analysis helps evaluate the viability of launching an artificial intelligence powered documentation platform in an increasingly saturated, fast-evolving developer tools market.

2.4.1 Threat of New Entrants

The global developer tools market is projected to reach $19.5 billion by 2031, growing at an impressive 17% Compound Annual Growth Rate (Verified Market Research, 2023). This growth creates an appealing landscape for new players and heightens competitive pressure for early-stage startups. However, the technical complexity of building real-time, artificial intelligence powered documentation systems presents a substantial barrier to entry. While venture capitalists poured more than $27 billion into generative artificial intelligence startups in 2023 alone — a trend expected to accelerate — the bar remains high for platforms that require deep infrastructure integration and artificial intelligence explainability. As a result, the threat of new entrants for a specialized platform like CodeDocs is moderate: market interest is growing, but few newcomers are equipped to deliver at the technical depth required.

2.4.2 Bargaining Power of Buyers

Buyers in the developer tools ecosystem — individual developers, engineering teams, and DevOps leads — wield considerable bargaining power. There is a wide availability of free or low-cost alternatives for documentation, including Markdown, Notion, GitHub Wikis, and Confluence. Price sensitivity is particularly acute among startups and freelance developers, who are unlikely to adopt a paid solution unless it clearly demonstrates a return on investment. Additionally, enterprise buyers have raised their standards significantly. Over 82% of organizations now expect developer tool vendors to offer System and Organization Controls 2 compliance, Security Assertion

doi:10.6342/NTU202502429

Markup Language integration, and self-hosting deployment options (Invimatic, 2023). The developer community is also highly peer-influenced; negative reviews, poor onboarding, or lack of transparency can spread quickly and discourage adoption (Stack Overflow, 2023). For CodeDocs to succeed, it must not only demonstrate technical superiority but also communicate value and trust clearly to a highly discerning user base.

2.4.3 Bargaining Power of Suppliers

The supplier landscape for developer infrastructure is favorable to startups. CodeDocs depends on widely available platforms such as Git, VSCode, and cloud providers like Amazon Web Services and Google Cloud Platform. These suppliers offer well-documented Application Programming Interfaces, competitive pricing, and widespread compatibility, reducing reliance on any single vendor. The commoditization of infrastructure-as-a-service (IaaS) has lowered switching costs, making supplier power relatively weak (Verified Market Research, 2023). Additionally, many early-stage startups benefit from generous cloud credits — often exceeding $100,000 — from venture partners and accelerators (Chapel, 2020). This flexibility allows CodeDocs to avoid lock-in and optimize for performance and cost across providers. Even Integrated Development Environments like VSCode and IntelliJ now provide seamless plugin support with no licensing barriers (JetBrains, 2023). Supplier bargaining power, therefore, does not pose a strategic risk to CodeDocs.

2.4.4 Threat of Substitutes

Substitute products represent a significant threat in this market. Many engineering teams still rely on informal and low-cost tools like Markdown files, GitHub Wikis, and Notion pages to handle internal documentation. These tools offer flexibility and familiarity at little to no cost. According to the 2023 Stack Overflow Developer

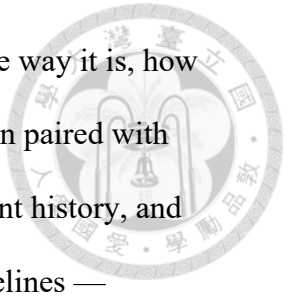
doi:10.6342/NTU202502429

Survey, Markdown remains the most widely used documentation format for codebases, reflecting developers' comfort with decentralized, lightweight alternatives.

The emergence of generative artificial intelligence tools has further intensified this threat. Platforms like GitHub Copilot and Tabnine provide real-time code suggestions, autocomplete functionality, and inline explanations — features that dramatically reduce the friction developers face when interacting with unfamiliar code. This evolution in developer tooling challenges the perceived value of traditional documentation platforms, which are often static, siloed, and decoupled from day-to-day workflows. GitHub Copilot surpassed 20 million users in 2025 (Lunden, 2025), underscoring the speed and scale at which artificial intelligence-based developer tools are being adopted across the industry. For CodeDocs, this signal both a risk and an opportunity. The risk lies in market perception: if artificial intelligence assistants are seen as "good enough," developer teams may deprioritize comprehensive documentation efforts or resist adopting new tools that appear redundant. However, it also creates a clear opportunity to position CodeDocs as a complementary — not competing — layer within the modern engineering stack. Rather than replacing code suggestions, CodeDocs is designed to surface context, institutional knowledge, and system-level understanding that artificial intelligence assistants cannot reliably reconstruct, especially in legacy or microservice-heavy codebases. This includes features like artificial intelligence powered codebase traversal, Guidebook Markers that persist team decisions and architectural rationale, and integrated documentation tied directly to pull requests, tests, and code owners.

To maintain relevance and win over users already embedded in the generative artificial intelligence ecosystem, CodeDocs must deliver clear and defensible differentiation. That means going beyond mere syntax-level commentary and offering

doi:10.6342/NTU202502429



intelligent infrastructure for understanding why the code is written the way it is, how modules interconnect, and what decisions were made over time. When paired with seamless collaboration workflows — such as in-line reviews, comment history, and integration with Continuous Integration/Continuous Deployment pipelines — CodeDocs becomes more than just documentation: it becomes an operational memory layer for the engineering team.

2.4.5 Industry Rivalry

Competitive intensity in the developer tools space is high. Established players such as Atlassian dominate with products like Jira, used by 57.5% of developers globally (Stack Overflow Developer Survey, 2024). These incumbents benefit from strong brand loyalty, enterprise contracts, and broad integration ecosystems. On the other end of the spectrum, open-source documentation tools have been adopted in over 3.9 million projects worldwide, with JavaScript frameworks leading usage (Statista, 2024). These projects often offer sufficient functionality for smaller teams at zero cost, placing pressure on proprietary solutions.

The rise of artificial intelligence native platforms further increases rivalry. Tools like GitHub Copilot are not just augmenting coding workflows — they are reshaping how engineers think about documentation itself. As generative models become more advanced, there is growing speculation about whether documentation and even some development tasks could eventually be handled entirely by artificial intelligence. While it’s unlikely that engineers will become obsolete, the demand for more intelligent, fewer manual solutions is accelerating. For CodeDocs to stand out in this environment, it must emphasize its unique value: deep explainability, real-time documentation accuracy, team-level collaboration features, and seamless integration with existing engineering

doi:10.6342/NTU202502429

tools. These differentiators are critical for building lasting defensibility and brand equity in a crowded and fast-moving space.

doi:10.6342/NTU202502429

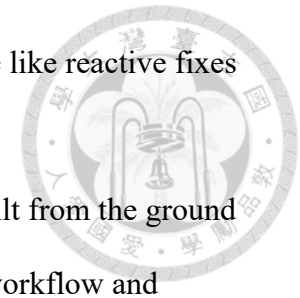
Table 3 Porter's Five Forces Analysis Summary for CodeDocs

Force	Impact Level	Summary
Threat of New Entrants	Moderate	The developer tools market is growing rapidly and attracting investment, but high technical barriers — including real-time code integration and artificial intelligence explainability — make it difficult for most startups to compete.
Bargaining Power of Buyers	High	Developers have many free or low-cost alternatives (e.g., Markdown, Notion, GitHub), and high expectations for compliance (e.g., System and Organization Controls 2). Peer reviews and product transparency heavily influence adoption.
Bargaining Power of Suppliers	Low	CodeDocs relies on commoditized infrastructure (e.g., Git, VSCode, cloud providers). Easy integration, competition among suppliers, and cloud credits minimize pricing pressure and vendor lock-in.
Threat of Substitutes	High	Lightweight tools and artificial intelligence-based coding assistants (e.g., Copilot) are widely adopted and reduce the perceived need for documentation. CodeDocs must offer distinct value in artificial intelligence context and team collaboration.
Industry Rivalry	High	The market is highly competitive with incumbents like Atlassian and emerging artificial intelligence tools reshaping developer behavior. CodeDocs must differentiate on usability, explainability, and workflow integration.

Note: Ratings are based on qualitative assessment of current market trends, developer tooling landscape, and cited academic and industry sources (e.g., Stack Overflow, Verified Market Research).

The software documentation market remains highly fragmented, with teams often patching together multiple tools to manage internal knowledge. CodeDocs differentiates itself by solving a core, often overlooked pain point: helping developers visualize the actual execution order of code—especially when logic is spread across disparate files and folders. Competing platforms such as Swimm and Mintlify lack this depth of functionality. Their solutions offer surface-level documentation support but require developers to exit their workflow and manually trace logic, often relying on

doi:10.6342/NTU202502429



incomplete or outdated annotations. As a result, these tools act more like reactive fixes than foundational solutions.

CodeDocs takes a fundamentally different approach. It is built from the ground up as a bottom-up system, integrating directly into the developer's workflow and mapping how code behaves in real time. Its traversal engine is not a bolt-on feature — it is the core of the platform's architecture. For competitors to replicate it, they would need to entirely re-engineer their systems, which are currently designed around passive, post-hoc documentation.

While companies like Swimm ($33.3M raised from Insight Partners), Mintlify ($125K from Y Combinator), and Codex ($4.5M, also Y Combinator-backed) have gained early attention, they remain in preliminary phases of development. CodeDocs, by contrast, offers a production-ready platform that directly addresses the developer pain of fragmented documentation and cognitive overload. By combining real-time traversal, artificial intelligence driven explainability, and seamless Integrated Development Environment integration, CodeDocs is positioned to become the foundational layer for internal engineering knowledge — not just a documentation add-on.

doi:10.6342/NTU202502429

	CodeDocs	Swimm.io	UseCodex.com	AI solution
Follows code execution to create documentation	✓	✗ Manually linked and created	✗ Only allows comments on code sections	✗ Function based comment generation (easily copied)
Quickly get answers to questions	✓	✗ None to date	✓ A user leaves a comment for another engineer to see	✗
Step by step guides	✓	✓	✗	✗
Multi repo functionality	✓	– Yes, but manually linked and created	✗ Comments only exist within one repo and aren't interconnected	✗ Function based comment generation
Frictionless Integration	✓	✗ Requires its own text editor	✗	✓
Generate wiki-like documentation for entire codebases & teams	✓	✗	✗	✗

Figure #2. Competitor Analysis of Feature Agains Codedocs

2.5 Strengths, Weaknesses, Opportunities, Threats Analysis

2.5.1 Strengths

CodeDocs is built on a strong technical foundation, with its Chief Technology Officer holding four patents in code observability — a clear indicator of deep technical expertise. The platform's proprietary bottom-up architecture, anchored by its traversal engine, allows developers to explore code execution without switching between platforms — an advantage that sets it apart from traditional documentation tools. In addition, the founding team has access to seasoned mentors and advisors from Silicon Valley, many of whom have facilitated early conversations with enterprise buyers. This level of access is uncommon at such an early stage and strengthens the company's go-to-market positioning. There is also growing interest among large organizations in self-hosted developer tools, particularly as enterprises look to maintain control over infrastructure and meet compliance requirements. CodeDocs has already secured a cohort of pre-committed beta users, aided by strong founder-network relationships and early-stage trust-building efforts.

doi:10.6342/NTU202502429

2.5.2 Weaknesses

As an early-stage startup, CodeDocs still faces several internal limitations. While the artificial intelligence powered documentation layer is part of the long-term product roadmap, its full implementation is projected for the second year after launch. Accelerating this timeline will require identifying and onboarding a highly specialized artificial intelligence developer. Brand awareness also remains limited; despite strong credentials, CodeDocs has yet to establish broad visibility among developers or enterprise buyers. Additionally, the team is heavily reliant on a few early technical hires, creating operational risk if any of those contributors were to leave. Lastly, the product has not undergone extensive user experience testing with non-technical stakeholders, which may limit adoption in cross-functional environments like product, compliance, or customer success teams.

2.5.3 Opportunities

The market for developer tools has undergone a significant transformation. As demand for software development has grown and workloads have become more exacting, developers — or perhaps their employers — appear to have become more receptive to spending on apps that boost productivity and save time (MacDonald, 2019). Today, it is one of the fastest-growing verticals in Software as a Service, driven by rising engineering headcounts and the increasing complexity of modern software systems. Code documentation remains a persistent pain point across all engineering teams — especially in distributed or remote-first organizations. As technology leaders focus more heavily on efficiency, onboarding, and institutional knowledge retention, tools that can preserve and enhance codebase literacy will play a central role. There is also growing demand for artificial intelligence integrated developer tools, as teams seek platforms that can not only document but also explain, refactor, and validate code

doi:10.6342/NTU202502429

autonomously. CodeDocs is well-positioned to serve this next wave of intelligent tooling.

2.5.4 Threats

The code documentation landscape is growing rapidly. In the past year, three new startups have entered the market with dedicated funding to solve documentation-related challenges. These competitors have secured early traction, validated the size of the opportunity but also intensified the race for mindshare and technical differentiation. Capital-raising will be essential for CodeDocs to accelerate hiring, complete security audits, and reach enterprise-grade readiness. Additionally, the platform's reliance on external ecosystems — such as GitHub and VSCode — introduces potential risks if integration terms or Application Programming Interfaces change. Finally, the rapid evolution of artificial intelligence powered developer tools means CodeDocs must continue to innovate aggressively to avoid being outpaced by newer entrants or incumbents expanding into this space.

2.6 Business Model Canvas

Sections 2.3–2.5 applied Political, Economic, Social, Technological, Legal, and Environmental, Five Forces and Strengths, Weaknesses, Opportunities, Threats lenses to surface regulatory tightening around code security, accelerating artificial intelligence adoption in developer tooling, moderate supplier power (commoditized cloud), and differentiating internal strengths (proprietary traversal architecture, early cohort access). Figure X translates these diagnostic insights into a coherent Business Model Canvas describing how CodeDocs creates, delivers, and captures value. The Customer Segments concentrate on venture-backed mid-market engineering organizations (20–100 developers) and regulated enterprises whose compliance constraints heighten willingness to pay, while an open-source/free tier seeds grassroots adoption. The Value

doi:10.6342/NTU202502429

Propositions—an always-current, execution-aware internal knowledge layer, measurable onboarding time reduction, and a self-host / air-gapped deployment path—directly mitigate pain points identified in interviews. Distribution Channels (open-source release, developer communities, accelerator networks, targeted outbound) support a product-led motion that shapes Customer Relationships around self-serve activation plus high-touch onboarding for regulated users. Revenue Streams (tiered Software as a Service + implementation and future premium artificial intelligence packs) are enabled by Key Resources (traversal engine intellectual property, seed capital runway, advisor leverage) and Key Activities (model refinement, security certification, community cultivation). Key Partnerships extend reach and enterprise credibility. A deliberately lean Cost Structure (engineering-heavy, controlled go-to-market, margin-protective inference costs) underpins modeled progression toward break-even and positive operating leverage.

doi:10.6342/NTU202502429

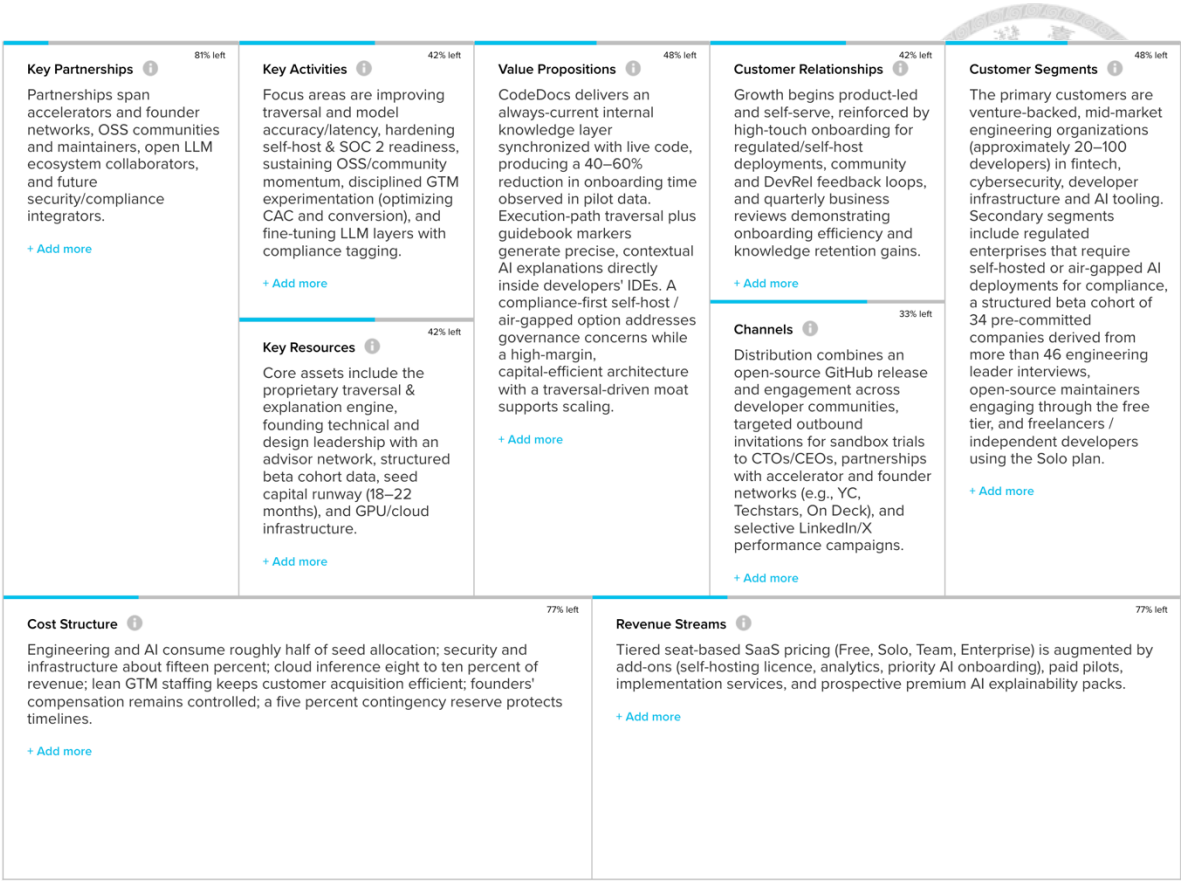


Figure #3. Business Model Canvas

3 Business Strategy

3.1 Product-Market Fit & Evolution

CodeDocs was initially conceived as a developer-first documentation platform that enhances knowledge sharing through real-time traversal and inline annotations. When the platform was first developed, the documentation pain point was acute — engineers lacked a seamless way to trace execution paths, and documentation tools were mostly external, static, or retrofitted. CodeDocs addressed this gap by embedding into the development workflow and making documentation native to the codebase.

However, the pace of innovation in developer tooling has accelerated dramatically, especially with the rise of generative artificial intelligence and large language model-based assistants. Tools like GitHub Copilot, Sourcegraph, and Tabnine are not only helping developers write code — they are increasingly explaining it. These

doi:10.6342/NTU202502429

models offer real-time code suggestions, inline explanations, and even documentation generation. As a result, CodeDocs in its current form risks becoming obsolete unless it evolves to match new expectations.

The strategic response is to reposition CodeDocs as an artificial intelligence native internal knowledge infrastructure tool — not just a documentation platform. The next iteration of the product must center around artificial intelligence assisted code explainability, compliance-tagged documentation, and team-specific onboarding intelligence. Rather than merely surfacing markdown or comments, CodeDocs would become an internal language model-powered layer that understands code history, function behavior, and tribal knowledge — fine-tuned to the unique structure and tone of each engineering team's codebase.

By training Large Language Models on internal repositories (via secure, self-hosted deployments), CodeDocs could offer natural language explanations, tailored onboarding workflows, and automatic documentation that stays up to date with every commit. This positions the platform as a defensible, infrastructure-level product that complements — not competes with — tools like Copilot by focusing on team-based institutional memory and system-wide knowledge traceability.

3.2 Placement Strategy

Given the rapid commoditization of public-facing artificial intelligence developer tools, CodeDocs' competitive advantage lies in its ability to serve a niche but growing segment: companies that cannot or will not use externally hosted artificial intelligence services due to compliance, security, or governance concerns.

Highly regulated industries — such as finance, government, defense, and healthcare — often prohibit developers from using cloud-hosted artificial intelligence tools or sharing sensitive code with third-party large language models. These

doi:10.6342/NTU202502429

organizations need self-hosted, on-premises artificial intelligence solutions that can operate in air-gapped or restricted environments. CodeDocs is uniquely positioned to address this gap.

By modifying the platform to support plug-and-play Large Language Model adapters, CodeDocs can integrate with internal artificial intelligence systems (e.g., fine-tuned open-source models like Mistral hosted on secure infrastructure). This allows the platform to deliver natural language explanations, smart onboarding, and context-aware documentation entirely within the customer's firewall — with no data ever leaving their network.

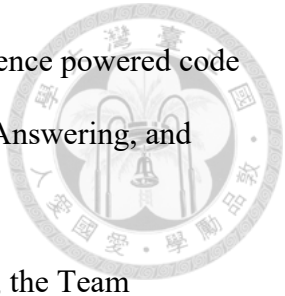
This approach not only addresses a real and underserved pain point, but also protects CodeDocs from direct competition with mass-market tools like Copilot. By shifting its placement strategy toward secure, enterprise artificial intelligence deployments, CodeDocs becomes not just a developer tool — but a compliance-first knowledge layer essential to scaling and securing engineering productivity within sensitive sectors.

3.3 Pricing Structure

CodeDocs adopts a tiered Software as a Service pricing model tailored to different user segments, ranging from open-source developers to enterprise teams. The Free (Open-Source Tier) is designed for individual developers or small teams working on public code. This plan supports one public repository and up to ten users, with the option to purchase additional seats at $10 per user per month. It includes access to CodeDocs' core traversal engine and external documentation hosting, enabling users to experience the platform's capabilities without incurring upfront costs.

The Solo (Private Repos) plan, priced at $15 per user per month, targets freelancers and independent developers working on private projects. This tier includes

doi:10.6342/NTU202502429



access to up to five repositories and features such as artificial intelligence powered code traversal, natural language explanations, collaborative Question and Answering, and external documentation hosting.

For growing teams or organizations with more advanced needs, the Team (Growth Tier) is available at $29 per user per month. This plan offers unlimited repository access and includes enterprise-oriented features such as Single Sign-On/Security Assertion Markup Language integration, audit logs, and administrative dashboards. Teams can deploy CodeDocs in either cloud or self-hosted environments, with the platform System and Organization Controls 2–ready for enterprise compliance. The plan also includes dedicated onboarding support, a private Slack or Discord channel, and a full Application Programming Interface access.

In addition to these core plans, CodeDocs offers optional add-ons such as a self-hosted license for the Team and Enterprise tiers, advanced analytics dashboards tailored to engineering leads, and priority artificial intelligence support for onboarding large and complex codebases.

doi:10.6342/NTU202502429

Table 4 Software Service Plans

Plan	Target Users	Pricing	Key Features
Free (Open Source)	Individual developers or small teams working on public code	Free (first 10 users), $10/user/month for additional seats	1 public repo, 10 users, CodeDocs core engine, external docs hosting
Solo (Private Repos)	Freelancers or independent developers with private projects	$15/user/month	Up to 5 repos, artificial intelligence powered traversal, team Q&A, external docs
Team (Growth Tier)	Larger teams with compliance or self-hosting needs	$29/user/month	Unlimited repos, Single Sign On/Security Assertion Markup Language, audit logs, admin tools, on-prem deployment, support
Optional Add-ons	Applicable to Team and Enterprise tiers	Custom pricing	Self-hosted license, analytics dashboards, priority artificial intelligence onboarding support

Note: Pricing is structured to encourage bottom-up adoption, with free access for public projects and scalable tiers for private and enterprise use.

4 Organization Management

CodeDocs operates as a lean, early-stage startup with a flat organizational structure designed to maximize agility, technical output, and fast decision-making. The founding team consists of a Chief Executive Officer (Victor Vong), Chief Operating Officer (Albert Lai), Chief Technology Officer (Mike Bartoli), and Head of Design (Illyn Chen). Each founder owns a distinct operational domain—engineering, operations/fundraising, product/tech infrastructure, and design, respectively—while maintaining close collaboration across all functions.

As a technical-first company, the core engineering team reports directly to the Chief Technology Officer, while future growth in sales, marketing, and customer success will fall under the Chief Operating Officer's purview. Design, user experience,

doi:10.6342/NTU202502429

and product feedback loops are centralized under the Head of Design, who interfaces directly with both engineering and early users.

The company is currently organized for rapid prototyping and market testing, with plans to scale the team functionally as product-market fit is validated. In the next 12–18 months, CodeDocs expects to hire dedicated roles in customer success, sales engineering, and DevOps, with a focus on supporting enterprise clients and self-hosted deployments.

doi:10.6342/NTU202502429

Table 5 Roles and Responsibilities

Role	Responsibilities	Reporting To
Chief Executive Officer	Product vision, strategic leadership, investor relations	Board of Directors
Chief Operating Officer	Operations, finance, business development, fundraising	Chief Executive Officer
Chief Technology Officer	Platform architecture, engineering leadership, artificial intelligence integration	Chief Executive Officer
Head of Design	User experience, product design, interface feedback	Chief Executive Officer
Engineering Team	Builds core platform features, reports to Chief Technology Officer	Chief Technology Officer
Design Team	User Experience implementation, reports to Head of Design	Head of Design
Future Hires: Sales & Customer Success	Manages client onboarding, support, and enterprise relationships	Chief Operating Officer
Future Hires: DevOps & Security	Ensures secure, scalable infrastructure for on-prem/self-hosted clients	Chief Technology Officer

Note: Titles reflect current functional responsibilities. Additional hires and role adjustments are planned as the team scales.

4.1 Key Team Members

The CodeDocs founding team brings deep expertise in engineering, venture strategy, and product design. Victor Vong, the Chief Executive Officer, holds a B.A. in Computer Science from University California of Berkeley and brings over five years of startup experience from companies such as Aether, Mashgin, Si14, and Solvvy. Most recently, he served as a Staff Software Engineer and Tech Lead at Zoom's Data Science

doi:10.6342/NTU202502429

team. Victor is known for his leadership in scaling technical teams and driving early product development.

Albert Lai, Chief Operating Officer, is a partner at MPTE Ventures where he has backed notable companies like Athelas, OpenSea, TalkDesk, and Veev Homes. He is a serial entrepreneur with multiple exits and currently advises early-stage companies in Software as a Service and infrastructure. Albert combines strategic investment insight with operational leadership.

Mike Bartoli, the Chief Technology Officer, previously held engineering roles at Solvvy, Salesforce, Defense Advanced Research Projects Agency, and Eli Lilly, and served as a Staff Software Engineer at Zoom. He holds four patents in observability and systems infrastructure and specializes in building scalable systems for enterprise-grade applications.

Illyn Chen, Head of Design, brings a creative background in illustration and interface design. Her work spans both consumer and enterprise products, and she is known for translating complex developer workflows into elegant, intuitive user experiences.

4.2 Equity Ownership Structure

Following the $2 million seed financing round at a $13 million pre-money valuation, CodeDocs adopted a founder-friendly equity structure that balances long-term incentive alignment with investor expectations and future hiring flexibility. Post-financing, seed investors hold 20% of the company, while 10% is allocated to an employee stock option pool to attract key hires. The remaining 70% is held by the founding team, distributed based on function, contribution, and long-term commitment.

The founding Chief Executive Officer retains 25% of the company and is responsible for vision, strategy, and capital raising. Both the Chief Technology Officer

doi:10.6342/NTU202502429

and Chief Operating Officer hold 20% each. The Chief Technology Officer oversees platform architecture, engineering leadership, and artificial intelligence infrastructure development. The Chief Operating Officer leads operations, go-to-market execution, and early customer success. A 5% stake is granted to the design partner who played a pivotal advisory role during the pre-launch product phase.

This structure ensures that key leadership roles are sufficiently incentivized, while maintaining operational flexibility and competitive hiring capability for future engineering and sales expansion.

Table 6 Equity Overview

Stakeholder	Ownership (%)	Post-Money Rationale
Founding Chief Executive Officer	25%	Leads vision, capital raising, and company strategy
Chief Technology Officer	20%	Owns engineering execution, infrastructure, and product architecture
Chief Operating Officer	20%	Over sees operations, growth, and go-to-market execution
Design Partner	5%	Provided early product and user experience feedback pre-launch
Employee Option Pool	10%	Reserved for engineers, sales, and future executive hires
Seed Investors	20%	Invested $2M at $13M pre-money valuation

Note: Equity distribution reflects initial contributions and long-term commitment. Allocations are designed to remain flexible as the company grows and key hires are onboarded.

4.3 Advisors and Mentors

CodeDocs is guided by a group of highly experienced advisors. Jonathan Goldick, a founding team member of Microsoft Word, also serves as Chair of the Carnegie

doi:10.6342/NTU202502429

Mellon Computer Science Advisory Board. He has advised companies in healthcare and software and was previously Entrepreneur-in-Residence at Healthcare Partners.

Patrick Yang, an advisor to CodeDocs, is the Co-founder and Managing Partner of Amity Ventures, where he raised over $500 million and built one of the top-performing global venture funds. A former investor at Highland Capital Partners, Patrick currently serves on the boards of several high-growth startups and brings deep insight into scaling software platforms.

5 Marketing and Sales Strategy

5.1 Audience & Positioning

CodeDocs is designed for engineering-centric software organizations, where the internal documentation burden is high and team efficiency is critical. Our primary users are software developers—technical professionals who engage with documentation tools daily. However, the actual purchasing decisions are made by Chief Technology Officers and Chief Executive Officers, depending on the organization's size and structure. Understanding and correctly targeting both groups is essential to our strategy.

Developers are highly influential in tool adoption but have unique behavioral patterns. They tend to avoid sales pitches, are generally skeptical of traditional advertising, and almost never respond to unsolicited outreach (Korop, 2017). Years spent working at their computers have given them a sharp filter for poor marketing (Kammerer, 2022). However, they are deeply embedded in online technical communities, frequently share resources, and trust tools recommended by their peers. This means the best way to reach developers is not through conventional channels, but through participation in authentic, technical spaces such as GitHub, Hacker News, and Reddit. Developers are also keen evaluators of tooling—they will use something that

doi:10.6342/NTU202502429

works well, is lightweight to integrate, and is open for inspection or contribution. For them, credibility comes from utility and transparency, not flashy branding.

On the other hand, Chief Technology Officers care about team velocity, onboarding speed, code maintainability, and reducing friction between product and engineering. They are often the gatekeepers for new developer tools. In many startups, especially those with growing engineering teams, the Chief Technology Officer must justify any tool that touches the codebase in terms of integration complexity, security, and long-term value. Messaging to this group must therefore emphasize the practical impact on engineering productivity, the ease of onboarding new developers, and the system's compatibility with existing workflows.

Chief Executive Officers, especially those at fast-scaling or venture-backed companies, tend to be focused on team efficiency, cost savings, and speed to market. They approve budgets for teamwide tooling and are responsible for balancing investments in engineering infrastructure with product development. While they are not technical buyers, Chief Executive Officers will often rely on input from their Chief Technology Officers while still needing to see a clear return on investment. For this audience, we emphasize time saved per engineer, lower onboarding costs, and reduced human error in product documentation workflows.

The companies we are targeting fall within high-growth, engineering-driven sectors such as fintech, cybersecurity, developer infrastructure, and artificial intelligence tooling. These organizations typically employ 20 to 100 engineers, operate with remote or hybrid teams, and are backed by institutional investors such as Sequoia, a16z, or Index Ventures. Specific examples of ideal early adopters include Vanta (security and compliance automation), Retool (developer tools), Ramp (expense management), Temporal (workflow orchestration), and Tailscale (network

doi:10.6342/NTU202502429

infrastructure). These companies have the budget to pay for high-leverage engineering tools, the technical leadership to understand our product's value, and the network visibility to influence similar firms in their space.

By targeting both the users (engineers) and the buyers (Chief Technology Officers and Chief Executive Officers) with clear, segmented messaging, CodeDocs positions itself as a tool that solves real pain for those building and scaling complex software systems.

5.2 Go-to-Market Execution & Cost

The go-to-market strategy for CodeDocs is built around a hybrid approach that combines community-based, bottom-up adoption with direct outreach to technical and business decision-makers. This approach reflects the real-world buying dynamics in modern software companies: while engineers initiate tool exploration, budget holders must ultimately approve purchases. Traditional marketing efforts such as Search Engine Optimization, Meta ads, and YouTube tend to create friction in this sales pipeline, as developers are often resistant to commercial-style promotions (Kammerer, 2022).

Our first channel for adoption is organic community engagement. We will open-source the core engine of CodeDocs, making it freely accessible on GitHub. This builds credibility and reduces adoption friction, allowing developers to try the product without engaging with a salesperson. We will also remain active on platforms where developers share knowledge—specifically Hacker News, Reddit, Stack Overflow, and Discord groups related to developer productivity. Our communication in these forums will avoid self-promotion and instead focus on surfacing real problems and inviting open discussion.

While community activity builds developer interest, direct outreach is essential for converting organizational buyers. Our outbound communication will focus on Chief

doi:10.6342/NTU202502429

Technology Officers and Chief Executive Officers at target companies. These messages will be personalized, and insight driven. Once engaged, prospects will be invited to try the tool with their teams in a sandbox environment.

From a cost perspective, we plan to invest in a lean but focused go-to-market team. This includes one full-time developer advocate responsible for open-source engagement, community credibility, and technical content, as well as one sales representative focused on outbound messaging to founders, Chief Technology Officers, and Vice Principles of Engineering. In addition, we anticipate a modest spend on tooling (email outreach platforms, customer relationship management systems, GitHub sponsorships, and light paid distribution on LinkedIn for founder-targeted content.

Our total marketing and sales budget is projected at $12,000 per month. This includes salaries, platform costs, and content support. We estimate a customer acquisition cost of under $1,200 per paying engineering team. At an annual contract value of $5,000 per customer, this Customer Acquisition Cost provides a healthy payback ratio and strong early-stage unit economics. More importantly, because early users are often deeply networked in the startup ecosystem, each successful customer has the potential to generate indirect referrals across peer companies.

In short, our go-to-market execution is built for sustainability and reach. It leans into the strengths of developer culture—credibility, transparency, and utility—while supporting it with a targeted, founder-facing outreach strategy to unlock decision-maker buy-in.

5.3 Early Traction, Influencers, and Advertising Strategy

After conducting interviews with over 46 staff engineers and Chief Technology Officers from early-stage and growth-stage startups, CodeDocs has secured commitments from 34 companies to participate in the upcoming beta test. These

doi:10.6342/NTU202502429

companies span the developer tooling, fintech, and cybersecurity sectors, and include engineering teams from Y Combinator-backed startups, as well as product-driven companies scaling their internal documentation processes. While names are under non-disclosure agreements until launch, a sample of participating Chief Technology Officers includes leaders formerly from Asana, Plaid, Cockroach Labs, and Linear—each with a strong track record of adopting tools that improve engineering velocity. This group will provide detailed usage feedback and serve as the basis for public case studies that showcase CodeDocs in live production environments.

Among our early supporters is the Chief Executive Officer of a top-ranking video conferencing software platform (name withheld due to contractual confidentiality), who expressed strong interest in embedding CodeDocs into their engineering onboarding flow. In his words, “The most effective way for CodeDocs to market its product is by allowing users to test the traversal engine with a single click. Imagine installing a plugin in your code editor, and instead of manually searching through files to understand how a line of code works, the tool instantly explains how that line is executed across the entire codebase—powered by artificial intelligence.” This emphasis on user experience and immediacy will guide how we showcase the product, particularly in developer demos, launch announcements, and community rollouts.

We also plan to formalize partnerships with top accelerators such as Y Combinator, Techstars, and On Deck. These ecosystems serve as powerful amplification networks, and CodeDocs will offer specially discounted startup packages to help companies in these programs quickly onboard engineering teams and reduce time-to-productivity. These partnerships also provide long-term brand equity, as many founders and engineers move across multiple startups and bring preferred tools with them.

doi:10.6342/NTU202502429

Our advertising strategy is tightly aligned with our product-led growth model. We will avoid high-cost, broad-based advertising channels that are historically ineffective for developer tools (Korop, 2017; Kammerer, 2022). Instead, we will focus on sponsoring technical newsletters, community podcasts, and open-source projects with overlapping audiences. Paid distribution will be limited to highly targeted campaigns on LinkedIn and Twitter, designed to reach Chief Technology Officers, Vice Principle of Engineering, and startup founders through posts about onboarding speed, developer velocity, and product delivery. These ads will be paired with gated product tours, allowing users to experience the traversal engine and document-sync feature without speaking to a sales rep.

By combining credible early supporters, real-time usage feedback from top technical leaders, and highly focused advertising channels, CodeDocs aims to drive sustained adoption from both grassroots developer communities and executive decision-makers.

6 Financial Management

6.1 Funding Request & Financial Projections

CodeDocs is seeking $2 million in seed round financing at a $13 million pre-money valuation. This capital is intended to support 18 to 22 months of operation, providing the company with sufficient runway to build its core technology, reach product-market fit, and convert early traction into recurring revenue. The round will be structured as a priced equity round, with capital allocated primarily toward product development, go-to-market execution, and regulatory readiness. The goal is to achieve between $2 and $3 million in annual recurring revenue within 22 months of funding, driven by a product-led growth strategy targeting technical teams at fast-scaling startups and mid-market enterprises.

doi:10.6342/NTU202502429

The financial model for CodeDocs is based on assumptions typical of high-growth B2B Software as a Service startups in the developer tooling space. It incorporates known benchmarks such as customer acquisition cost, lifetime value, and churn metrics seen in developer-focused Software as a Service platforms. Each subsection below provides a detailed breakdown of expected capital allocation, cash flow management, and growth trajectories under conservative operating assumptions.

6.2 Use of Funds

The $2 million seed round will be allocated across five core areas, with an additional strategic reserve set aside for flexibility.

doi:10.6342/NTU202502429

Table 7 Capital Efficient Operating Plan Overview

Category	Allocation United State Dollar	Percentage	Description
Product & Engineering Development	1,000,000	50%	Build the traversal engine, expand the artificial intelligence documentation layer, and develop user feedback loops.
Go-to-Market & Community Growth	40,0000	20%	Hire developer advocates, produce technical content, and drive early user conversion.
Security & Infrastructure	300,000	15%	Achieve System and Organization Controls 2 compliance and launch self-hosted deployments to satisfy enterprise requirements.
Operations & Administration	200,000	10%	Cover legal, accounting, payroll, cloud services, and other core operational expenses.
Contingency & Strategic Reserve	100,000	5%	Maintain a flexible reserve for runway extension or unplanned product and go-to-market needs.

Note: This structure reflects the capital-efficient operating plan of a deep-technology startup: development-heavy in the early stages, while remaining lean on customer acquisition and operational overhead.

6.3 Financial Projections Overview

The financial model for CodeDocs includes projections across the first three years of operation, beginning with the close of the seed round in Q1 2025. These projections were developed using a bottom-up forecasting model based on expected user growth, team hiring schedules, and realistic per-team pricing assumptions. The full model includes four key components:

1. **Startup Cost Summary** - A clear breakdown of how initial capital is allocated across engineering, security, go to market, and operational support.

doi:10.6342/NTU202502429

2. **Income Statement Forecast (2025–2027)** - A multi-year projection of revenue, gross margin, operating expenses, and net earnings, including Earnings Before Interest, Taxes, Depreciation, and Amortization and net profit margin calculations.
3. **Balance Sheet Forecast** - A snapshot of expected cash reserves, receivables, liabilities, and equity over time, highlighting capital efficiency and asset-light operations.
4. **Break-even Analysis** - Breakeven was estimated using contribution margin, burn-to-revenue pacing, and scenario-based growth modeling. This yielded a practical timeline for achieving profitability.

6.4 Startup Cost Summary

CodeDocs's upfront capital requirements are structured to reflect the company's product-led approach and high technical development needs. Most startup costs are allocated toward engineering talent and infrastructure, with smaller but essential budgets for go-to-market motion, regulatory readiness, and administrative stability. These allocations are based on industry benchmarks for early-stage Software as a Service companies building deeply integrated developer tools.

doi:10.6342/NTU202502429

Table 8 Initial Fund Deployment Following Seed Funding

Category	Estimated Cost (USD)	Description
Backend Engineering Hires	700,000	Recruiting and onboarding a high-caliber team of 4–5 full-stack and artificial intelligence engineers.
Artificial Intelligence Infrastructure & Model Tuning	150,000	Compute credits, fine-tuning, and scaling context-aware documentation models.
Security & Self-Hosting Setup	150,000	Implementing System and Organization Controls 2 requirements and building isolated deployments for enterprise clients.
Developer Advocacy & Community	100,000	Hiring advocates, producing technical launch content, and supporting early user feedback.
Legal & Corporate Setup	50,000	Company formation, intellectual property registration, data protection compliance, and employment contracts.
Administrative Tools & Software	25,000	Setting up payroll systems, Customer Relationship managements, cloud credits, and communication tools.
Marketing & Design Assets	25,000	Initial brand assets, user experience kits, landing page dev, and explainer videos.
Office/Remote Setup & Equipment	25,000	Equipment stipends for remote staff, software licenses, and remote work enablement tools.
Strategic Reserve	75,000	Financial buffer for timeline delays, contractor work, or user acquisition acceleration.

Note: Allocation is focused on product development, infrastructure, and go-to-market execution to support a 22-month runway and accelerate early adoption.

Total initial startup cost is around $1,300,000. These figures exclude ongoing monthly burn and are intended to cover the setup phase, spanning approximately the first 6–8 months before repeatable growth metrics and renewals begin to stabilize the business.

6.5 Cash Flow Overview

doi:10.6342/NTU202502429

CodeDocs projects a strong cash position over the first three years of operations, driven by disciplined cost control, a capital-efficient product roadmap, and steady revenue growth from paid pilots, subscriptions, and integrations.

In Year 1 (2025), the company plans to raise $2 million in seed funding, which will cover foundational team hires, core infrastructure, and early go-to-market activities. Projected incoming revenue of $500,000 and an additional $25,000 from pilot fees and grants bring the total incoming cash to $2.525 million. After operating and capital expenditures, the company expects to close the year with $660,000 in cash.

In Year 2 (2026), revenue is forecasted to rise to $2 million, with continued support from grants. Fixed and variable costs increased modestly, primarily to support growth in headcount, cloud infrastructure, and compute capacity. Despite no new equity funding, the ending cash balance is projected at $320,000.

By Year 3 (2027), the business scales significantly with anticipated revenue of $4.5 million and $100,000 in other income. Operating expenses expand proportionally, particularly in areas like model compute, DevOps, and product design. However, the projected cash position by year-end is strong at $1.765 million, reflecting CodeDocs' ability to self-fund growth without further dilution. This positive trajectory demonstrates financial resilience, operational leverage, and a path toward breakeven by late 2026 or early 2027.

6.5.1 Considerations (Narrative Elaboration)

The cash flow projections for CodeDocs were carefully modeled to balance conservative planning with high-growth assumptions typical of venture-backed Software as a Service startups. Each line item in the financial forecast reflects underlying assumptions about market adoption, operating needs, and capital efficiency, all of which are closely tied to strategic business goals outlined in this thesis. The Year

doi:10.6342/NTU202502429

1 projection is designed to prioritize foundational team building and infrastructure, while Years 2 and 3 anticipate scale-up following initial product-market fit and successful enterprise adoption.

6.5.2 Revenue Timing & Realization

The revenue model assumes that CodeDocs begins generating income in Year 1 through paid pilot programs and early enterprise contracts, targeting approximately $500,000 in gross revenue. This figure was derived based on pricing benchmarks from comparable Software as a Service developer tools, assuming 10 to 15 mid-market clients paying between $30,000 and $50,000 annually. In Year 2, revenue grows to $2,000,000, representing expanded sales capacity and stronger go-to-market execution. This figure accounts for a wider customer base, the beginning of team expansion in sales and marketing, and the compounding effects of renewals and upselling to existing clients. By Year 3, the model forecasts $4,500,000 in revenue, based on deeper market penetration, stronger sales operations, increased usage-based revenue, and partnerships. This stepwise growth reflects a standard enterprise sales cycle of 6 to 12 months, while also acknowledging that delays in closing strategic accounts or slower-than-expected pilot conversions may defer the realization of cash flows. To mitigate these timing risks, deferred revenue accounting and cautious burn multiples have been considered.

The “Other Income” line reflects anticipated non-recurring revenue such as small innovation grants and pilot program fees. In Year 1, this is estimated at $25,000, rising to $50,000 in Year 2 and $100,000 by Year 3. These figures are based on current programs offered by Taiwanese government agencies, startup accelerators, and strategic technology partnerships that reward early adoption or public-private collaboration.

doi:10.6342/NTU202502429

6.5.3 Cost Creep in Variable Infrastructure

The projections recognize that CodeDocs' infrastructure costs will increase proportionally with platform adoption, especially due to the compute demands of artificial intelligence powered code traversal and analysis. In Year 1, total cloud infrastructure and artificial intelligence and machine learning compute costs are estimated at $80,000. These figures are based on expected compute usage patterns for inference-heavy models operating on Amazon Web Services or Google Cloud Platform and benchmarked against pricing tiers for similar applications running inference workloads on managed services. In Year 2, as user load increases and data processing become more complex, these costs are projected to rise to $140,000, and by Year 3, the total reaches $280,000. This upward trend accounts for higher service level agreements, 24/7 uptime, and the need for increased redundancy as CodeDocs onboards enterprise customers. Nonlinear growth in infrastructure costs is expected unless offset by engineering efforts to optimize compute usage or adopt more cost-efficient architectures.

Merchant and payment processing fees are calculated as a percentage of gross revenue, following typical Software as a Service billing structures. In Year 1, these fees are projected at $320,000, based on a blended processing fee of approximately 6% of gross revenue. As revenues grow to $2,000,000 and $4,500,000 in Years 2 and 3 respectively, these fees are forecasted to rise accordingly to $470,000 and $670,000. These estimates consider volume discounts as well as fees associated with recurring billing platforms like Stripe or Chargebee.

6.5.4 Human Capital Scalability

Salaries and benefits are modeled to increase gradually as the company scales. In Year 1, total salary and benefit expenses are projected at $600,000. This reflects a

doi:10.6342/NTU202502429

core team of approximately six to eight full-time employees, including founders, engineers, and one or two generalist operators. These figures are benchmarked against average compensation rates for remote software teams, combining Taiwan-based employees and one or two U.S.-based hires. In Year 2, headcount is expected to increase to around 10 to 12 team members, with compensation scaling to $750,000. By Year 3, the headcount is projected to grow to approximately 15 to 18 members, bringing the total cost of salaries and benefits to $900,000. These estimates are designed to reflect responsible growth while maintaining a high talent bar.

In addition to full-time staff, the model includes budget allocations for contractor and freelancer contributions. In Year 1, this cost is projected at $30,000, which supports fractional contributors such as legal counsel, User Experience designers, or part-time marketers. This amount increases to $40,000 in Year 2 and $50,000 in Year 3, assuming increased usage of third-party experts for documentation audits, marketing assets, and technical advisory roles.

Founders' stipend or draw is projected at $360,000 annually for the first two years, assuming modest compensation shared across the founding team. By Year 3, this increases to $480,000 to reflect formalized executive compensation as the company scales, brings on external board members, and enters discussions for potential Series A fundraising or strategic partnerships.

6.5.5 Capital Efficiency & Optionality

The model intentionally avoids assuming future equity funding beyond the initial $2,000,000 seed round in Year 1. This capital infusion is intended to cover initial team building, product development, and go-to-market efforts through to breakeven. The resulting ending cash balance at the end of Year 3—projected at $1,765,000—demonstrates capital efficiency while preserving optionality for future funding rounds.

doi:10.6342/NTU202502429

The company may elect to raise a Series A to accelerate growth, enter new markets, or fund ecosystem integrations. Having a strong cash position improves CodeDocs' negotiating leverage in such scenarios and allows the team to be selective in future financing.

6.5.6 Macroeconomic and Market Dynamics

While developer infrastructure products are considered mission-critical, the model accounts for external risks that may affect procurement cycles. These include shifts in macroeconomic conditions, such as rising interest rates, tighter venture capital deployment, and cuts in Information Technology spending by large enterprises. These risks may delay contract signings or reduce average deal sizes. Although these variables are difficult to predict precisely, the model conservatively assumes a lag between product readiness and revenue acceleration, particularly in Year 1, to account for potential headwinds in enterprise procurement cycles.

6.5.7 Burn Rate Management

To maintain financial agility, the model emphasizes a disciplined approach to burn rate and cost management. The total operating expenses—combining fixed and variable costs—remain under control throughout the three-year period, rising only in proportion to revenue growth. In Year 1, the total burn (excluding capital expenditures) is $1,815,000. This increases to $2,350,000 in Year 2 and $3,125,000 in Year 3. These increases are justified by the corresponding growth in revenue and the deliberate scaling of headcount, infrastructure, and support. Capital expenditures are limited to $50,000 in Year 1 and decrease over time, reflecting limited investment in physical infrastructure due to the cloud-native and remote-first nature of the business. This allows CodeDocs to preserve optionality and extend runway while adjusting burn based on live key

doi:10.6342/NTU202502429

performance indicators such as monthly recurring revenue, customer acquisition cost, and cash-on-hand thresholds.

In summary, the financial projections presented in this model are designed to demonstrate CodeDocs' viability and scalability while retaining flexibility for future fundraising and operational pivots. The numbers are based on bottom-up modeling, market benchmarks, and conservative estimates where uncertainty exists. As the company scales, maintaining visibility into these metrics—and being ready to adjust based on performance—will be critical to long-term success.

doi:10.6342/NTU202502429

Figure 11 Projected Cash Flow Statement for CodeDocs (2025–2027)

Category	Year 1 (2025)	Year 2 (2026)	Year 3 (2027)
Starting Cash Balance	$0	$660,000	$320,000
Seed Round Funding	$2,000,000	$0	$0
Incoming Revenue	$500,000	$2,000,000	$4,500,000
Other Income (Pilot Fees, Grants)	$25,000	$50,000	$100,000
Total Incoming Cash	$2,525,000	$2,050,000	$4,600,000
Fixed Expenses			
└─ Salaries & Benefits	$600,000	$750,000	$900,000
└─ Office Rent (Remote infra)	$50,000	$60,000	$70,000
└─ Insurance & Legal	$40,000	$50,000	$60,000
└─ Accounting & Compliance	$30,000	$35,000	$40,000
└─ Advertising & Paid Media	$30,000	$35,000	$40,000
└─ Product Design	$150,000	$250,000	$350,000
└─ Software Subscriptions	$60,000	$85,000	$100,000
└─ Travel & Events	$45,000	$45,000	$45,000
└─ Founders Draw / Stipend	$360,000	$360,000	$480,000
Total Fixed Expenses	$1,365,000	$1,670,000	$2,085,000
Variable Expenses			
└─ Cloud Infrastructure	$20,000	$40,000	$80,000
└─ Artificial Intelligence Model Compute	$60,000	$100,000	$200,000
└─ Contractor Costs (Freelancers)	$30,000	$40,000	$50,000
└─ Support & DevOps	$10,000	$15,000	$20,000
└─ Sales Commissions	$10,000	$15,000	$20,000
└─ Merchant/Payment Fees	$320,000	$470,000	$670,000
Total Variable Expenses	$450,000	$680,000	$1,040,000
Fixed Expenses	$1,365,000	$1,670,000	$2,085,000
Variable Expenses	$450,000	$680,000	$1,040,000
Capital Expenditures	-$50,000	-$40,000	-$30,000
Ending Cash Balance	$660,000	$320,000	$1,765,000

doi:10.6342/NTU202502429

Note: Forecasted inflows and outflows based on revenue growth and seed capital deployment.

6.6 Income Statement Forecast

The income statement derived from CodeDocs' cash-flow assumptions portrays a company moving rapidly from early-stage losses to strong operating profitability. Revenue rises from $525 thousand in 2025 to $4.7 million by 2027, a compound annual growth rate of roughly 240 percent. Cost of goods sold—principally cloud infrastructure and artificial intelligence model inference—tracks at approximately eight percent of revenue, delivering a gross margin that remains steady around ninety percent. This margin profile is typical of capital-light, cloud-native Software as a Service platforms and confirms that artificial intelligence compute expenses have been sized realistically rather than optimistically.

Operating expenses increase at a far slower rate than revenue. Head-count-driven line items such as salaries, developer advocacy, and customer success expand in absolute dollars but fall as a percentage of revenue, creating operating leverage. The result is an Earnings Before Interest, Taxes, Depreciation, and Amortization swing from a loss of $1.29 million in 2025 to a positive $2.44 million in 2027, with Earnings Before Interest, Taxes, Depreciation, and Amortization margin improving from –246 percent to fifty-one percent over the horizon. After a nominal depreciation charge—representing the straight-line amortization of modest capital expenditures—net income turns positive in 2027 at $1.94 million, yielding a net profit margin of forty-one percent. Once federal and local taxes are factored at an assumed twenty percent rate, after-tax earnings still exceed forty per cent of revenue, underscoring the scalability of the model.

doi:10.6342/NTU202502429

6.6.1 Considerations

Sustaining high gross margins depends on disciplined management of cloud and inference costs. Should user-generated content grow faster than expected, artificial intelligence compute spend could outpace revenue unless usage caps or model-compression strategies are enforced. On the revenue side, projections assume consistent conversion of pilot companies into paid subscriptions and limited churn; slower enterprise adoption would delay Earnings Before Interest, Taxes, Depreciation, and Amortization break-even and compress net margins. Salary inflation in key engineering markets represents another variable—salary bands have already returned to pre-layoff levels in 2024, and further competition for artificial intelligence talent could lift fixed expenses.

Finally, the model treats capital expenditures as minimal, in line with a remote-first, cloud-hosted architecture. Any strategic decision to operate proprietary inference clusters would materially lift CapEx and depreciation, moderating free-cash generation in later years. Management should therefore revisit infrastructure strategy annually, balancing gross-margin preservation with performance requirements.

Overall, the statement depicts a financially resilient path to profitability, but its success depends on strict cost governance, continued product-market fit, and disciplined reinvestment of early cash surpluses into growth levers that maintain operating leverage.

doi:10.6342/NTU202502429

Table 9 Projected income statement from 2025 through 2027

Category	2025	2026	2027
Revenue	525000	2050000	4700000
Cost of Goods Sold	-80000	-140000	-280000
Gross Profit	445000	1910000	4420000
Operating Expenses	-1735000	-2250000	-1985000
Earnings Before Interest, Taxes, Depreciation, and Amortization	-1290000	-340000	2435000
Depreciation & Amortization	-10000	-10000	-10000
Operating Income Earnings Before Interest, Taxes, Depreciation, and Amortization	-1300000	-350000	2425000
Interest Expense	0	0	0
Income Before Tax	-1300000	-350000	2425000
Income Tax	0	0	-485000
Net Income / (Loss)	-1300000	-350000	1940000
Gross Margin (%)	84.8%	93.2%	94.0%
Earnings Before Interest, Taxes, Depreciation, and Amortization Margin (%)	-245.7%	-16.6%	51.8%
Operating Margin (%)	-247.6%	-17.1%	51.6%
Net Margin (%)	-247.6%	-17.1%	41.3%

Note: Projections are based on conservative growth estimates, with assumptions around developer adoption, enterprise conversions, and gradual increase in average revenue per user.

6.7 Balance Sheet Forecast

The projected balance sheet (Table 10) confirms that CodeDocs maintains a conservative capital structure while steadily converting operating gains into liquidity and equity value.

doi:10.6342/NTU202502429

6.7.1 Asset profile

Cash and equivalents rise from $0 in January 2025 to $1.77 million by December 2027, mirroring the positive ending-cash balances shown in the cash-flow schedule. Accounts-receivable growth—from $25 000 to $480 000—tracks roughly one month of sales, reflecting 30-day enterprise payment terms without stressing working capital. Capitalized software and equipment reach just $100 000 by 2027, consistent with a cloud-native, remote-first operating model and the modest CapEx detailed in the cash-flow statement.

6.7.2 Liability structure

Current liabilities remain light. Accounts payable equate to about one month of operating costs, while deferred revenue climbs to $350 000 as multi-year contracts are billed up front. The tax accrual of $85 000 in 2027 matches 20 percent of projected pre-tax profit for that year. No long-term debt is assumed, keeping the company unlevered and interest-expense-free.

6.7.3 ***Equity movement***

Paid-in capital stays fixed at the initial $2 million seed raise. Retained earnings evolve directly from the income-statement net-income line: cumulative losses of $1.65 million over 2025–26 is fully offset by the $1.94 million profit in 2027, leaving positive retained earnings of $185 000. Total shareholders' equity therefore expands from $595 000 in 2025 to $2.19 million by 2027, supporting a healthy assets-to-equity ratio of roughly 1.1×.

6.7.4 Balance-Sheet Considerations

1. **Liquidity Cushion** - The 2027 cash balance of $1.77 million provides approximately nine months of forward operating coverage at projected burn rates,

doi:10.6342/NTU202502429

giving management flexibility to accelerate research and development or defer a Series A until market conditions are favorable.

2. **Deferred-Revenue Obligation** - Up-front billings improve cash, but associated service obligations must be delivered over the contract term. Accurate revenue recognition and customer success resourcing are essential to avoid performance liabilities.
3. **Working-Capital Discipline** - Conservative receivables (≈30 days) and payables (≈30 days) minimize liquidity strain. If larger enterprise accounts negotiate 60- to 90-day terms, the company may require additional working-capital financing or tighter collections.
4. **CapEx Versus OpEx** - The forecast assumes minimal on-prem inference infrastructure. If CodeDocs elects to host proprietary Large Language Models for data-residency–sensitive customers, CapEx and depreciation could materially rise, lowering return on assets and operating cash.
5. **No Long-Term Debt** - An unlevered balance sheet reduces financial risk but limits tax shields. Management should revisit the capital structure before scaling internationally, where selective debt could be cheaper than equity.
6. **Equity Optionality for Growth** - With positive retained earnings by 2027, CodeDocs can raise a Series A from a position of strength or self-fund further expansion. Maintaining high cash-conversion efficiency will be critical to preserve this leverage in negotiations.

Together, these factors underscore a balance sheet that supports disciplined growth while maintaining financial resilience—key to navigating competitive pressures in the rapidly evolving artificial intelligence developer-tooling market.

doi:10.6342/NTU202502429

Table 10 Projected balance sheet from 2025 to 2027

Category	2025	2026	2027
Cash & Equivalents	660000	320000	1765000
Accounts Receivable	25000	180000	480000
Pre-paid & Other Current Assets	10000	25000	40000
Capitalized Software & Equipment (net)	50000	80000	100000
Total Assets	745000	605000	2385000
Accounts Payable	80000	110000	190000
Deferred Revenue	50000	150000	350000
Accrued Expenses & Taxes	20000	25000	85000
Total Liabilities	150000	285000	625000
Seed-Capital Equity	2000000	2000000	2000000
Retained Earnings	-1405000	-1755000	185000
Total Equity	595000	245000	2185000
Total Liabilities + Equity	745000	605000	2385000

Note: Table reflect anticipated capital structure and asset growth based on seed funding, controlled burn rate, and reinvestment strategy during the initial scale-up phase.

6.8 Break-even Highlight

The cash flow forecast reflects CodeDocs's capital allocation strategy following the successful close of a $2 million seed round, which is included as the company's opening cash balance in Year 1. This capital supports the first 18 to 22 months of operations and is fully accounted for in the beginning of the forecast timeline.

The company generates modest revenue in Year 1 from early subscription contracts and paid pilots. Despite incurring an operating loss in the first year, cash flow remains positive due to conservative spending across fixed and variable costs. Key expenses include backend development, cloud infrastructure, user experience design, legal compliance, and strategic developer advocacy.

doi:10.6342/NTU202502429

With limited capital expenditures and a lean go-to-market approach, CodeDocs ends Year 1 with a healthy cash reserve. By Year 2, the company approaches operational profitability and continues scaling customer adoption. Year 3 reflects strong revenue growth, improved operating efficiency, and a resulting cash balance exceeding $4 million—sufficient to continue scaling without immediate follow-on financing.

6.9 Break-even Analysis

The break-even table demonstrates that CodeDocs remains loss-making through its first two operating years but crosses decisively into profitability during 2027. Revenue in 2025 is intentionally modest—derived mainly from pilot licenses and early subscriptions—and covers less than one-third of the fixed expense base, yielding a negative Earnings Before Interest, Taxes, Depreciation, and Amortization of roughly 1.3 million dollars. In 2026 revenues more than triple, yet operating losses persist because fixed infrastructure, payroll expansion and go-to-market outlays still exceed the contribution generated by sales. The pivotal change occurs in 2027: once cumulative revenue passes roughly 2.2 million dollars, the 92-percent contribution margin fully offsets fixed costs, so every incremental sale thereafter flows almost entirely to the bottom line. By December 2027 the business is expected to post 2.44 million dollars in Earnings Before Interest, Taxes, Depreciation, and Amortization and 1.94 million dollars in net income, all while finishing the year with 1.77 million dollars in cash—evidence that profitability and liquidity inflect simultaneously.

6.9.1 Break Even Considerations

The break-even timeline is sensitive to three principal variables. First, revenue realization must materialize on schedule; slips in enterprise-contract start dates or slower pilot conversions could defer profitability by a full fiscal year. Second, the contribution-margin ratio assumes cloud-infrastructure and artificial intelligence

doi:10.6342/NTU202502429

compute costs remain below ten percent of revenue; unexpected spikes in large-language-model pricing or inefficient inference usage would raise the break-even bar. Third, fixed spending is projected to flatten after 2026, banking on operational leverage; unplanned head-count additions or security-compliance requirements could erode the cost discipline that underpins this projection. Continuous monitoring of these levers—sales velocity, gross-margin integrity, and cost discipline—will be essential to ensure the break-even point remains achievable within the 2027 calendar year.

Table 11 Projected Break-Even Analysis

Metric	2025	2026	2027
Revenue	525000	2050000	4700000
Cost of Goods Sold	-80000	-140000	-280000
Contribution Margin (~92%)	445000	1910000	4420000
Fixed & Semi-Fixed Op-Ex	-1740000	-2250000	-1985000
Earnings Before Interest, Taxes, Depreciation, and Amortization	-1290000	-340000	2435000
Break-Even Revenue Needed	1890000	2450000	2160000

Note: Break-even is reached mid-2027 when cumulative revenue overtakes fixed costs, after which each additional sale flows largely to profit.

7 Uncertainty Risks and Future Plans

While CodeDocs introduces a novel traversal engine that enhances artificial intelligence assisted documentation, the innovation also exposes the company to market timing and strategic replication risks. The core premise behind CodeDocs is that artificial intelligence systems, even the most advanced, cannot efficiently explain large-scale codebases due to their inability to follow execution paths without reprocessing the

doi:10.6342/NTU202502429

entire codebase repeatedly. CodeDocs solves this by providing an execution-aware traversal layer that serves as the structural backbone for more accurate and scalable documentation.

However, the company's advantage is not unassailable. The most immediate risk is that larger incumbents—such as GitHub Copilot, Amazon CodeWhisperer, or OpenAI—could build similar traversal functionality directly into their models or Integrated Development Environments. If that occurs before CodeDocs secures sufficient user adoption or enterprise integration, its differentiation could erode rapidly. There is also the possibility that general-purpose artificial intelligence models will improve to a point where traversal-like behavior is approximated through brute force compute, reducing CodeDocs's engine to a performance boost rather than a category-defining advantage.

This risk is compounded by macro-level trends in software engineering. In 2024, Salesforce publicly stated that it would halt new engineering hires in 2025 due to increased productivity from artificial intelligence coding tools (Martin, 2024). Chief Executive Officers Marc Benioff cited a 30% efficiency gain from internal artificial intelligence systems. If this pattern holds across the industry, the overall market size for developer tools could shrink, especially those targeting code generation or review workflows.

Despite these concerns, CodeDocs remains positioned in a layer of the stack that is not easily replaced. Unlike prompt-driven code generation tools, CodeDocs focuses on comprehension—helping developers, especially in large teams, understand code they did not write. As code becomes more auto generated and complex, the need for clarity, traceability, and auditability increases. CodeDocs meets this demand by enabling smaller teams to manage larger and more opaque codebases confidently.

doi:10.6342/NTU202502429

The risk to CodeDocs, therefore, is not technical feasibility—it is the pace at which others attempt to replicate traversal, combined with the macro contraction of developer roles. If the company cannot move fast enough to establish a usage moat or convert early traction into paid adoption, it may face cash flow pressure, shortened runway, or lower valuations in subsequent rounds. CodeDocs's long-term defensibility relies on delivering its plugin ecosystem quickly and embedding deeply into the developer workflow before traversal becomes commoditized.

doi:10.6342/NTU202502429

Reference

Bankiya, Sunil, and Valerio Pozzato. “Developers Are the New Rockstars of SaaS: You Either Sell through or to Them.” *Medium*, 8 Mar. 2021, https://medium.com/dawn-capital/developers-are-the-new-rockstars-of-saas-you-either-sell-through-or-to-them-9933e54904aa.

Chapel, Jay. “11 Ways to Get AWS Credits.” *Medium*, 24 Nov. 2020, https://jaychapel.medium.com/11-ways-to-get-aws-credits-22a01793d38.

Dror, Tzvi. “Engineer Onboarding: The Ugly Truth about Ramp-Up Time.” *HackerNoon*, 26 Sept. 2020, https://hackernoon.com/engineer-onboarding-the-ugly-truth-about-ramp-up-time-7e323t9j.

European Commission. *General Data Protection Regulation (GDPR) Compliance Guidelines*, https://gdpr.eu/ (accessed May. 18, 2023).

Geerligs, Rick. “How Many Developers Are in the US and in the World [Updated].” *Daxx Software Development Teams*, 23 Sept. 2021, https://www.daxx.com/blog/development-trends/number-software-developers-world.

Invimatic. *SOC 2 Compliance: The Complete Guide for SaaS*. 2023, https://www.invimatic.com/blog/soc-2-readiness-assessment-guide/.

JetBrains. *The State of Developer Ecosystem 2023*. 2023, https://www.jetbrains.com/lp/devecosystem-2023/.

Johnson, Tom. “Council Post: The Real Problem with Tech Professionals—High Turnover.” *Forbes*, 10 Dec. 2021, https://www.forbes.com/sites/forbesbusinessdevelopmentcouncil/2018/06/29/the-real-problem-with-tech-professionals-high-turnover.

doi:10.6342/NTU202502429

Johnson, Tom. “Why Documenting Code Is So Difficult.” *Documenting APIs*, 8 Oct. 2021, https://idratherbewriting.com/learnapidoc/docapis_code_difficulty.html.

Lunden, Ingrid. “GitHub Copilot Crosses 20 Million All-Time Users.” *TechCrunch*, 30 July 2025, https://techcrunch.com/2025/07/30/github-copilot-crosses-20-million-all-time-users/.

Kammerer, Max. “The Code to Advertising to Developers.” *BuySellAds Content Portal*, 16 Nov. 2022, https://content.buysellads.com/advertisers/how-to-advertise-to-developers.

KPMG International. *Key Global Trends in Sustainability Reporting*. 2022, https://home.kpmg/xx/en/home/insights/2022/09/survey-of-sustainability-reporting-2022/global-trends.html.

MacDonald, Jesse. “Council Post: Developer Tools Quietly Become a Growth Story.” *Forbes*, 18 Oct. 2019, https://www.forbes.com/sites/forbestechcouncil/2019/10/16/developer-tools-quietly-become-a-growth-story.

Martin, Hayden. “Salesforce Will Hire No More Software Engineers in 2025, Says Marc Benioff.” *Salesforce Ben*, 18 Dec. 2024, https://www.salesforceben.com/salesforce-will-hire-no-more-software-engineers-in-2025-says-marc-benioff.

Mordor Intelligence. “Software Development Tools Market Size & Share Analysis (2025 – 2030).” Mordor Intelligence, 2025, https://www.mordorintelligence.com/industry-reports/software-development-tools-market.

Rachel, Taylor. “Why Documentation Matters, and Why You Should Include It in Your Code.” *freeCodeCamp.org*, 3 July 2019,

doi:10.6342/NTU202502429

https://www.freecodecamp.org/news/why-documentation-matters-and-why-you-should-include-it-in-your-code-41ef62dd5c2f.

Research, Insight. “Software Development Tools Market Size Booming at a CAGR of 14.5% During 2022–2027.” *GlobeNewswire*, 19 Apr. 2022, https://www.globenewswire.com/en/news-release/2022/04/19/2424149/0/en/Software-Development-Tools-Market-Size-Booming-at-a-CAGR-of-14-5-During-2022-2027-Extensive-Research-by-Industry-Size-Recent-Developments-Market-Position-Growth-Drivers-Business-Op.html.

Srivastava, Trehan, Wagle, Wang. *Developer Velocity: How Software Excellence Fuels Business Performance*. McKinsey, 2020, https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/developer-velocity-how-software-excellence-fuels-business-performance.

Stack Overflow. *2023 Developer Survey*. 2023, https://survey.stackoverflow.co/2023.

Stack Overflow. *2024 Developer Survey*. 2024, https://survey.stackoverflow.co/2024/.

Staffing Industry Analysts. “Tech, Retail and Media Sectors See Highest Turnover Rates: LinkedIn Research.” *Staffing Industry Analysts*, 2023, https://www.staffingindustry.com/news/global-daily-news/tech-retail-and-media-sectors-see-highest-turnover-rates-linkedin-research.Statista. *Total Number of Global Open Source Projects Adopted 2024*. 2024, https://www.statista.com/statistics/1419477/open-source-projects-adopted/Statista.

doi:10.6342/NTU202502429

Statista. *Total Number of Global Open Source Projects Adopted 2024*. 2024, https://www.statista.com/statistics/1419477/open-source-projects-adopted/Statista.

TechCrunch. *A Comprehensive List of 2025 Tech Layoffs*. 2025, https://techcrunch.com/2025/04/11/tech-layoffs-2025-list.

Verified Market Research. *Software Development Tools Market Size & Forecast*. 2023, https://www.verifiedmarketresearch.com/product/software-development-tools-market/.

Wolf, David. "Why Developers Write Horrible Documentation and How to Solve It." *DZone DevOps*, 13 July 2016, https://dzone.com/articles/why-developers-write-horrible-documentation-and-ho.

Woodgate, Robert. "What Is Markdown, and How Do You Use It?" *How-To Geek*, 11 Jan. 2022, https://www.howtogeek.com/448323/what-is-markdown-and-how-do-you-use-it.

World Intellectual Property Organization (WIPO). *Guide to Generative AI and Intellectual Property*. 2023, https://www.wipo.int/meetings/en/details.jsp?meeting_id=82908.

doi:10.6342/NTU202502429