OmniGrep: New Age Code Search and Visualisation

Tags	AI DevTools

Problem Statement

Software developers spend an average of **X hours per week** searching for information across codebases, documentation, and communication channels. This results in reduced productivity, context switching, and slower development cycles. The lack of **effective search and visualization tools** for codebases is a significant bottleneck, especially in large organizations where existing tooling is either fragmented or outdated.

This results in **30-60% productivity loss** during code comprehension activities, affecting the speed of issue resolution and feature development. The problem becomes more acute as teams scale and as more diverse technologies and languages are integrated into a codebase.

Validation Process

Interviews and Insights

- Conducted **67 interviews** with developers and tech leads across organizations ranging from startups to large enterprises.
- Spoke with the founder of DoWhile AI, who confirmed the pain point but indicated a degree of uncertainty on how solutions might evolve. There are competitors rushing to find the appropriate product for optimal long term PMF. See Competition Analysis for more information on how various solutions are positioned.
- A recurring theme was that developers **waste hours context switching** between different tools and sources, and there is no unified way to extract

the information they need for decision-making on **where to insert the new code that satisfies tests, requirements, and doesn't break previous code**.

- Key metrics:
  - **75% of interviewees** expressed dissatisfaction with current tools (e.g., SourceGraph) for deep context search.
  - Average **2-4 hours daily spent** on code search activities. **40% - 70%** of time taken, from ticket assignment to pull request merge is spent on understanding where and how various components interact. This includes classes, to modules, to complete microservices and how all of them interconnect. We believe that this **40-70% figure can be distilled to 5%-10%** with our solution hypothesis.

Risks

Market Risk

Software development might change drastically in the coming decade that invalidates looking at source code. A world where AI agents are doing most of the code writing shrinks the market signficantly. Positioning can be done such that OmniGrep is the **base layer code search engine,** that all other coding agents will utilise to understand and navigate the whole codebase. The complete AI Software Engineer dream seems to be about 3-5 years later, and we can position ourselves that way once we have market reputation.

Sales Risk

- **Trust Issues**: Companies may hesitate to give a startup, led by a relatively young founder, access to their codebases.
- **Positioning Strategy**: Need to create a **compelling narrative** around deep integration, value-add, and **data privacy/security** to overcome trust barriers. An open source GTM strategy, for competitions such as Google Summer of Code and Hacktoberfest, etc might be a great way to break into this domain and get reputation and initial design partners.

Primary Incumbent: SourceGraph

- SourceGraph started by copying internal developer tools found at major tech companies like Google and Facebook and making SaaS for other companies. They are now worth **2.6B$.** Their core focus is no longer on code search, but has shifted to Cody, their GitHub Copilot competitor.

Hypothesis: What happens at big tech often trickles down to others as a SaaS offering.
This has been seen with SourceGraph and Glean both. SourceGraph's CTO saw the code search internal tool at Google during his brief one year stint, and met his CEO at Palantir Technologies. Glean is a SaaS-ification of an internal tool known as MOMA, an internal search engine for all enterprise data.

We believe a similar thing is going to happen with another internal tool known as Duckie. This tool does exactly what our Solution Hypothesis (below) does, and has been a life saver for many L3-L5s at Google as the "go-to answer engine" for code and docs. This has been enabled only because LLMs and RAG have matured enough to hallucinate at low rates.

Initial Solution Hypothesis

- Build a **Retrieval-Augmented Generation (RAG)** model that pulls context from:
  - **Codebase**: Indexed and versioned for easy historical access.
  - **Issue Tracker**: Leverage GitHub/Jira issues to provide context around decisions.
  - **Tech Docs**: Ingest internal documentation and relevant external docs.
  - **QnA Channels**: Slack, Discord, or Teams where informal knowledge sharing occurs.
- Unify these sources into a single interface, which **reduces context switching** and provides more actionable insights directly within the developer workflow.

Additional Sections

Customer Personas

- **Software Engineers**: Need to quickly understand unfamiliar codebases.
- **Tech Leads**: Need visibility across multiple projects for effective coordination.
- **Engineering Managers**: Want to reduce cycle time and improve team productivity.

Intended Enterprise Customers: Series A and onwards engineering teams, with high feature request volumes that require extreme agility. Avoid large stable organisations that might build instead of buy for initial GTM, approach when market reputaton is impeccable.

Competitive Analysis

- **SourceGraph**: Strong in search but lacks comprehensive visualization and insights on cross-tool dependencies. Their core focus is now Cody (Copilot competitor) and not
- **Startups**
  - **Storia AI:** https://storia.ai/
  - **Greptile:** https://www.greptile.com/
    Have reached seed for 4.1M$ with Initialised Capital leading the round. Positioned as an API now for making internal developer tools such as Slack Experts, Issue enrichers, and coding agents.
  - **CodeViz:** https://www.codeviz.ai/
  - **GitHub Copilot Chat:** There is a chance GitHub will double down on "chat to codebase" along with visualisations as well. This will nip us in the bud.

Unique Selling Proposition (USP)

- **Unified RAG System**: Integrates multiple sources beyond just code, making it easy for developers to get all relevant context in one place.
- **Visualization Layer**: Providing graph-based or dependency-based visualizations for easier understanding.
- **Developer-first Focus**: Built with developer UX in mind—low friction, seamless integration with existing workflows.

- **Future Plan:** Imbue agency into OmniGrep and let it create initial Pull Requests and Issues from text prompts. This adapts the tool to the new AI Agents age and positions it as a AI Engineer as well.

Key Metrics for Success

- **Time Saved**: Reduction in time spent on search-related tasks by at least **30%**.
- **Adoption Rate**: Percentage of users using the tool daily vs. weekly.
- **Usage Frequency (Most Important)**: Code search is a daily problem, and our solution must be a daily use product.
- **Customer Satisfaction (NPS)**: Aiming for a Net Promoter Score above **50** during pilot phase.

Key Performance Indicators (KPIs)

- **Monthly Active Users (MAU)**: Track growth in active users of the platform.
- **Freemium Conversion Rate**: Percentage of free users converting to paid plans.
- **Community Engagement**: Number of contributions and forks in open source repositories.

Ideal Customer Profile (ICP)

- **Small to Mid-sized Development Teams**: Teams of **10-50 developers** who need better tooling to understand complex codebases, both the internal ones and the external dependencies that they utilise.
- **Open Source Projects**: Projects with distributed contributors who require a cohesive way to search and understand code.
- **High Growth Startups**: Startups scaling their engineering teams and struggling with onboarding and maintaining code comprehension.
- **DevOps and Site Reliability Engineers (SREs)**: Need to understand cross-system interactions and troubleshoot incidents quickly.

Go-to-Market (GTM) Strategy

- **Open Source Libraries**: Release core components as open source to build community trust and adoption.
- **Public Blogging**: Write detailed technical blogs about solving hard tech problems—targeting Hacker News, Reddit, and engineering blogs for visibility.
- **Developer Competitions and Events**: Participate in **Hacktoberfest**, **Google Summer of Code (GSoC)**, and other open source competitions like **Outreachy** and **MLH Fellowships** to onboard developers and raise awareness.
- **Partnerships**: Partner with open source communities and developer advocacy groups to increase reach and credibility. **Get enterprise design partners ASAP.**