USE BLOCKS

Open Source Doesn't Pay the Bills

But What Does?

An honest account of building a business on open source

Max Pabinger · CEO & Co-Founder, useblocks

US Community Day · 5th of November 2026

Intro

Raise your hand if you know this button

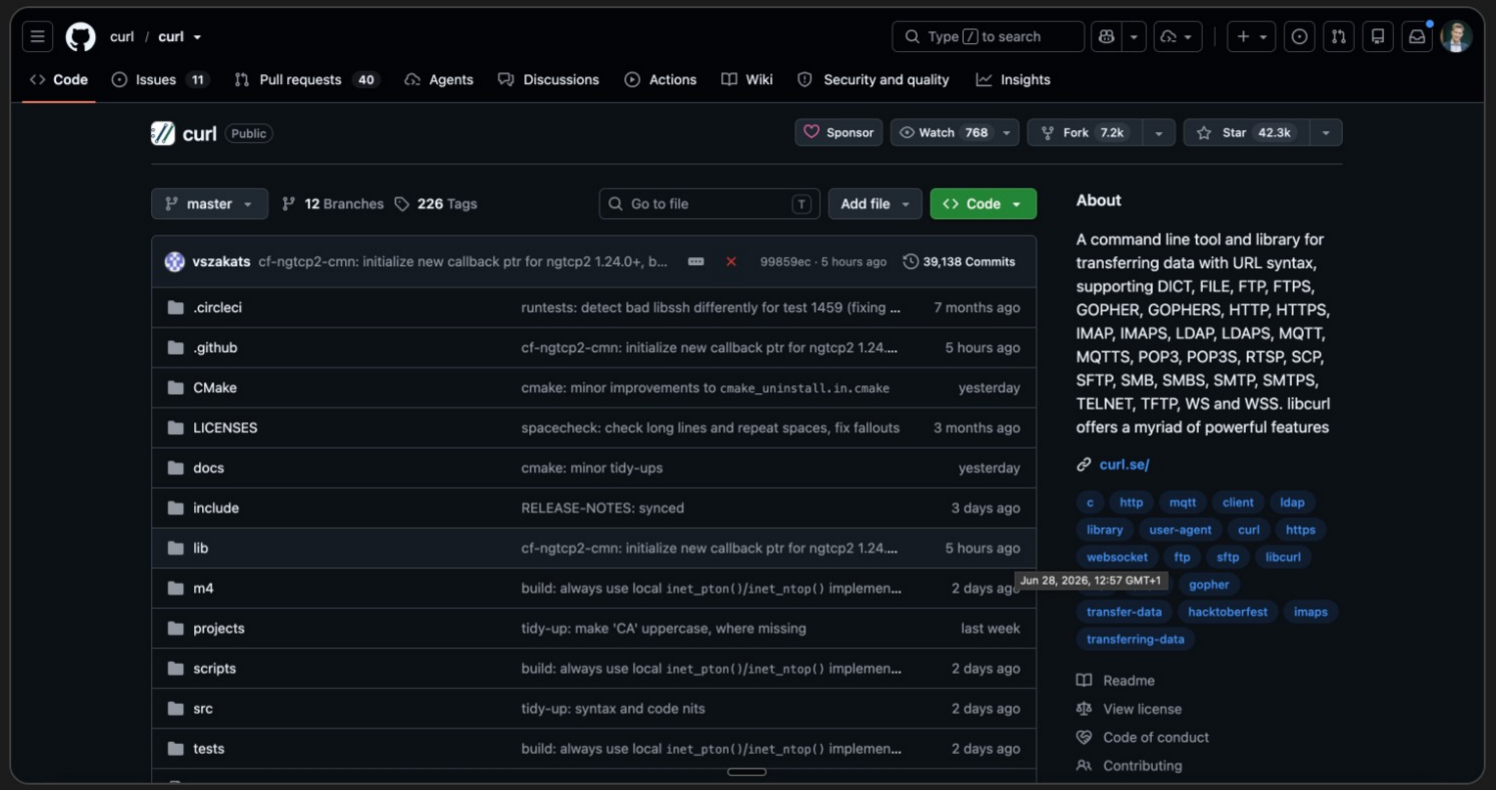
USE BLOCKS



github.com/sponsors

Intro

You already depend on projects that use it



curl — the tool inside every HTTP request you've ever made

Tools the whole industry relies on — **asking for donations.**

USE BLOCKS



Who has ever sponsored an open source project?

Raise your hand.

USE BLOCKS



Who has ever received money from the community — got paid *because* of open source?

USE BLOCKS

Open source is a collaboration model.
It is not a business model.

"Sponsor me" can feel a lot like **charity.**
So when open source doesn't pay the bills — **what does?**

Today's Story

Today: how useblocks tried to get paid for open source — and what finally worked

USE BLOCKS

1



Who we are

A startup behind Sphinx-Needs — 300K+ monthly downloads, born in automotive.

2



What we tried

Memberships, consulting, grants, procurement — most of it failed. Honestly.

3



What worked

Finding product-market fit a second time — on top of the open core.

The honest version. The failures included — because that's where the lessons are.

Who We Are

useblocks at a glance

USE BLOCKS

300K+

Monthly Downloads

Sphinx-Needs — the de facto standard for requirements-as-code

5M+

Total Downloads

Community traction has always been real

30

Team Members

Engineers, consultants & product experts around Munich

10+

Enterprise Customers

OEMs, Tier 1s & tech leaders running useblocks in production

72

Participants at Last UGM

Our User Group Meeting — up from 18 in 2024

Who We Are

It started 8 years ago — inside an OEM

USE BLOCKS

~2018 · ORIGIN



Born at BMW

Sphinx-Needs created to manage requirements as code — by engineers, for engineers.

→

SCALE-UP



Hardened at Bosch

Adopted and pushed further at scale — proving it in serious safety-critical work.

→

TODAY



Industry standard

300K+ monthly downloads. A community of safety & security engineers worldwide.

The open-source core was forged inside the exact industry it now serves.

Who We Are

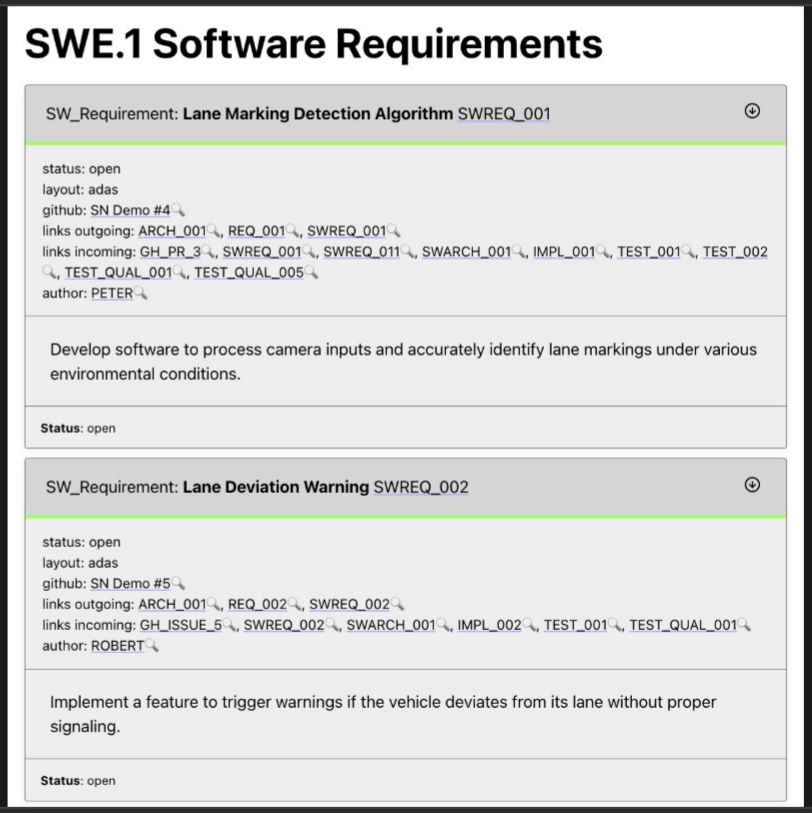
What is Sphinx-Needs? Requirements - as code, in Git

USE BLOCKS

SOURCE (.RST IN GIT)

```
4   .. _SWE1_Software_Requirements:
5
6   SWE.1 Software Requirements
7   ===========================
8
9   .. swreq:: Lane Marking Detection Algorithm
10     :id: SWREQ_001
11     :status: open
12     :links: ARCH_001, REQ_001
13     :author: PETER
14     :github: 4
15
16     Develop software to process camera inputs and
17     accurately identify lane markings under various
18     environmental conditions.
19
20  .. swreq:: Lane Deviation Warning
21     :id: SWREQ_002
22     :status: open
23     :links: ARCH_001, REQ_002
24     :author: ROBERT
25     :github: 5
26
27     Implement a feature to trigger warnings if the
28     vehicle deviates from its lane without proper
29     signaling.
```

RENDERED (SPHINX-NEEDS)



SWE.1 Software Requirements

SW_Requirement: **Lane Marking Detection Algorithm** SWREQ_001

status: open
layout: adas
github: SN Demo #4
links outgoing: ARCH_001, REQ_001, SWREQ_001
links incoming: GH_PR_3, SWREQ_001, SWREQ_011, SWARCH_001, IMPL_001, TEST_001, TEST_002, TEST_QUAL_001, TEST_QUAL_005
author: PETER

Develop software to process camera inputs and accurately identify lane markings under various environmental conditions.

Status: open

SW_Requirement: **Lane Deviation Warning** SWREQ_002

status: open
layout: adas
github: SN Demo #5
links outgoing: ARCH_001, REQ_002, SWREQ_002
links incoming: GH_ISSUE_5, SWREQ_002, SWARCH_001, IMPL_002, TEST_001, TEST_QUAL_001
author: ROBERT

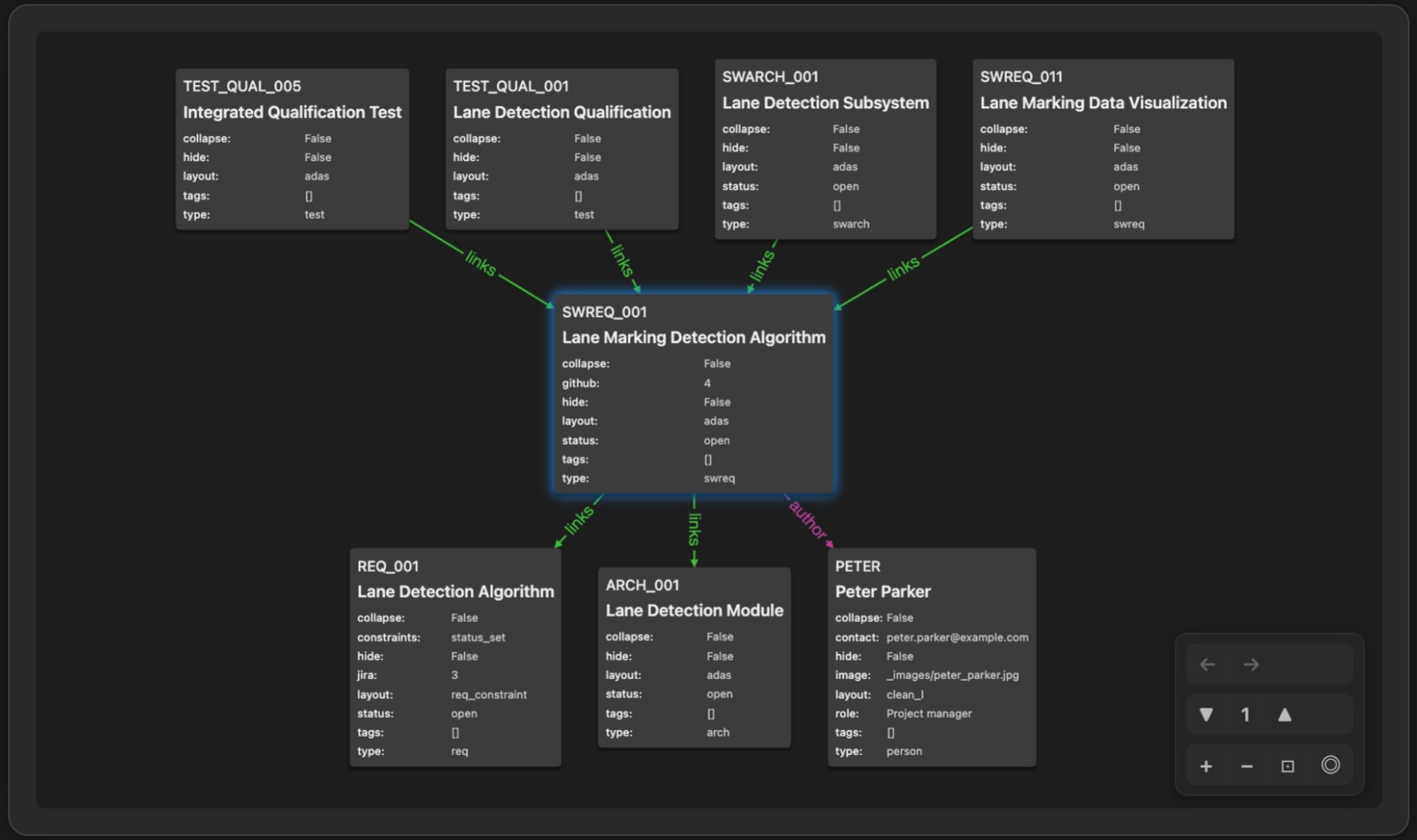
Implement a feature to trigger warnings if the vehicle deviates from its lane without proper signaling.

Status: open

How it works

All objects link - the traceability tree

USE BLOCKS



USE BLOCKS

So what does it actually take to build a business on open source?

The Thesis

You have to find Product-Market Fit — twice

USE BLOCKS

PMF #1 · OPEN SOURCE

Does the community love it?

A tool people adopt without being sold to. Downloads, stars, contributors, word of mouth.

✓ Solved — 300K downloads/month

→

PMF #2 · BUSINESS

Who pays — for what, and why now?

A paid layer of value on top of the free core. Revenue, not gratitude.

✗ The hard one. Took years.

Most open-source startups only ever solve the first. The gap between **downloads and dollars** is where they die.

The Thesis

A quick definition

USE BLOCKS

Product-market fit means being in a good market with a product that can **satisfy that market** — and the market **pulling product out of you.**

— after Marc Andreessen

PMF #1 · Open source

PMF #1 we already had: developers, not buyers

USE BLOCKS

Git-native

Requirements live where the code lives. Version-controlled, reviewable, diffable.

No tool switch

Engineers stay in their editor. No heavyweight ALM suite to log into.

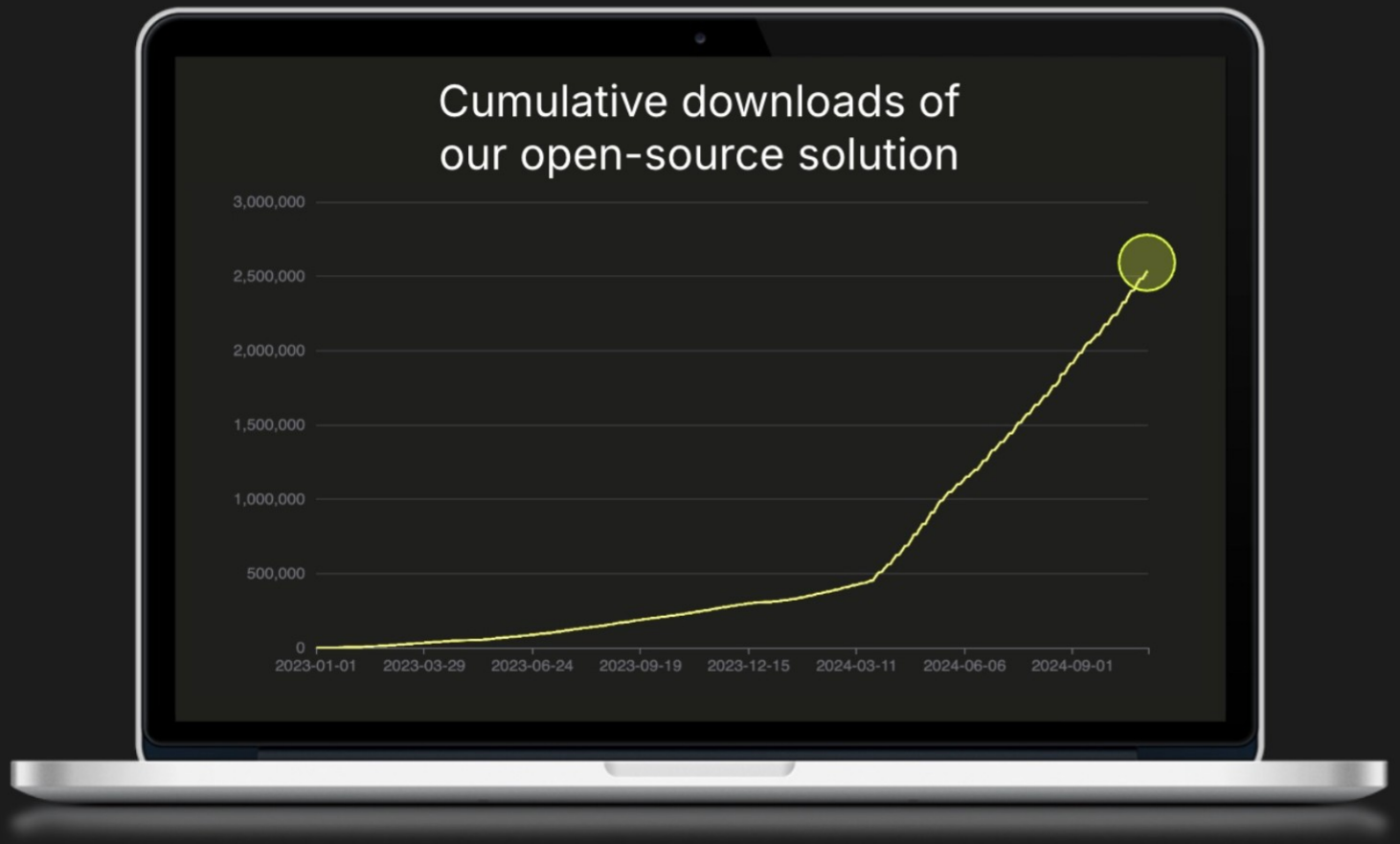
Built for the SDV

The software-defined vehicle treats everything as code — Sphinx-Needs fits natively.

> The community pull was real. But a developer who loves your tool is **not** the person who signs the cheque.

Traction

Sphinx-Needs has real traction in the market



341k
Downloads monthly

5M
Total Downloads

>30
Enterprise Users

USE BLOCKS

PMF #2 is harder — because you have to be worth paying for on top of something free

You're not selling against competitors.
You're selling against your own free product.

The Odds

The OSS commercialization funnel is brutal

USE BLOCKS

~630,000,000	GitHub repositories (raw activity)
~11,400,000	Actual OSS projects (published packages)
~800 (~0.007%)	Ever became VC-backed OSS companies
~100	Ever reached an exit (IPO / M&A) derived: 12% of ~800 ≈ 96
~40	Ever crossed $100M revenue was "<10" in 2013

Roughly: **1 in ~285,000** OSS projects becomes a $100M business.

Sources: GitHub Octoverse 2025 (itsfoss.com) · ecosyste.ms 2025 package dataset via Open Source Security interview (opensourcesecurity.io) · Serena Capital + Linux Foundation, State of Commercial Open Source 2025 (linuxfoundation.org PDF) · COSSI 2025 ($100M threshold trend, including "<10" in 2013) (coss.community).

The Landscape

How did others cross the gap?

USE BLOCKS

PROJECT	HOW MONEY IS MADE	THE PATH
Linux (kernel ecosystem)	Corporate maintainers funded by employers; monetized via enterprise distros, support, and integration services	**Foundation + enterprise services**
Kubernetes (CNCF)	Core project is free; money comes from managed platforms (EKS, GKE, AKS), enterprise tooling, and support	**Open standard + cloud services**
PostgreSQL	No single vendor owns it; companies monetize with managed DB, HA tooling, and enterprise support	**Ecosystem monetization**
GitHub	Acquired by Microsoft, 2018 — ~$7.5B	**Acquisition**
GitLab	Open-core + SaaS tiers + enterprise subscriptions	**Open-core subscriptions**
Docker	Developer subscriptions, Docker Hub plans, and enterprise products for teams	**Developer SaaS + enterprise**
Red Hat	Subscription & support on Linux → IBM, 2019, $34B	**Long game + acquisition**
Astral (uv, Ruff)	VC-funded (Accel, a16z) → acquired by OpenAI, Mar 2026	**VC money → acquisition**
Kùzu (graph DB)	Acquired by Apple, Oct 2025 — repo archived, no longer maintained	**Acqui-hire (project ends)**

Sources: official announcements & vendor materials (Linux Foundation, CNCF, PostgreSQL, GitHub/Microsoft, Red Hat/IBM, GitLab, Docker, Astral/OpenAI, Kùzu/Apple).

USE BLOCKS

1

What We Tried

Four honest attempts — and why most of them didn't pay the bills

What We Tried

The graveyard — and one survivor

USE BLOCKS

① OSS membership

for enterprises

✗ No urgency to pay

② Consulting on top

services revenue

△ Useful cash, but doesn't scale

③ Förderprogramme

Mod2SDV + AI4SDV

△ Good signal, limited business impact

④ Open core model

value on top + AI

✓ This one worked

Four headstones. One survivor. Let's walk through each.

What We Tried · ①

OSS membership for enterprises

USE BLOCKS

💡 THE IDEA

Companies that depend on Sphinx-Needs pay an annual membership — funding maintenance, support and a stable roadmap.

⚠️ WHY IT DIDN'T PAY

No perceived urgency. The tool already works for free. In cost-saving cycles, memberships were often the first budget line cut. "Why pay for something we're not blocked on?" Free-riding is rational.

What We Tried · ②

Consulting services on top

USE BLOCKS

💡 THE IDEA

Sell expertise — integration, training, custom setup. Real revenue, real customers, immediately.

⚠️ WHY IT DIDN'T PAY

It works — but it trades hours for euros. Revenue is capped by headcount. It doesn't scale, and it pulls the team away from product.

> Consulting kept the lights on. It was never going to build a company.

What We Tried · ③

Förderprogramme: Mod2SDV and AI4SDV

USE BLOCKS



Mod2SDV

Strong project signal from the automotive space.



AI4SDV

Clear AI relevance and ecosystem visibility.

KEY INSIGHT: **We truly appreciated collaborating with all partners, but these programs are built far more for enterprises than for startups.**

USE BLOCKS

2

What Finally Worked

PMF #2 — found on top of the open core, when AI changed the value equation and VC runway enabled deep product investments

Toolsuite

The full ecosystem: from code to compliance

USE BLOCKS

€

ubCode

Real-time validation, ontology checks & AI guardrails

€

ubConnect

Unify ALM tools into one traceable data pipeline

OSS

sphinx needs

Create, manage & validate structured needs

OSS

🔗 Codelinks

Trace code ↔ docs with a single comment

OSS

🧪 Test-Reports

Import test results as traceable need objects

€

ubTrace

Audit-ready traceability & compliance insights

€

OSS

🤖 Pharaoh

Agentic AI framework — authoring, tracing & compliance workflows across Input, Build and Analytics

Input **Build** **Analytics**

What Worked · Open Core Model

The enterprise layer customers actually pay for

USE BLOCKS

ubConnect

ubConnect

- **No rip-and-replace:** Integrates existing systems — DOORS, Jira, Polarion, ReqIF — bidirectionally into the intent layer.
- **Adoption bridge:** Teams keep their current ALM tools and investment while gaining x-as-code traceability. No migration risk.
- **The unlock:** Without ubConnect, enterprises with legacy tooling cannot adopt our stack. This removes the last blocker.

ubCode

ubCode

- **Speed:** Sphinx-Needs OSS builds take up to 1 hour on large projects. Our Rust engine delivers in under 500 ms — same project, real-time feedback as you type.
- **Quality:** Ontology rules enforced in real time. Linting catches structural errors before a single build runs.
- **AI-native:** CLI + MCP server makes spec-driven development with AI agents (SDD) practical and auditable.

ubTrace

ubTrace

- **Access for everyone:** Data analytics and governance for non-coders — quality officers, project leads and auditors work directly in the intent graph without touching code.
- **Live compliance:** Runs HARA, TARA, FMEA and other safety/security workflows on top of the living artifact graph.
- **Audit-ready by default:** Compliance evidence emerges automatically as a byproduct of daily work — not assembled in a last-minute scramble.

VC runway enabled this: The Rust rewrite, the MCP integration, the HARA/TARA workflows — none of this existed two years ago. Capital let us build ahead of the AI shift instead of catching up to it.

What Worked

Then AI changed the value equation

VC-backed investment enabled the Rust rewrite, live linting, and AI workflow integration when timing mattered.



← AI made this cheap

This got even more valuable →

AI made the coding part near-free.
The proof part — where we live — became the bottleneck.

USE BLOCKS

> In the age of AI coding agents,
> **intent is the new source code**,
> and **intent is the new type system**.

What Worked · The Toolsuite

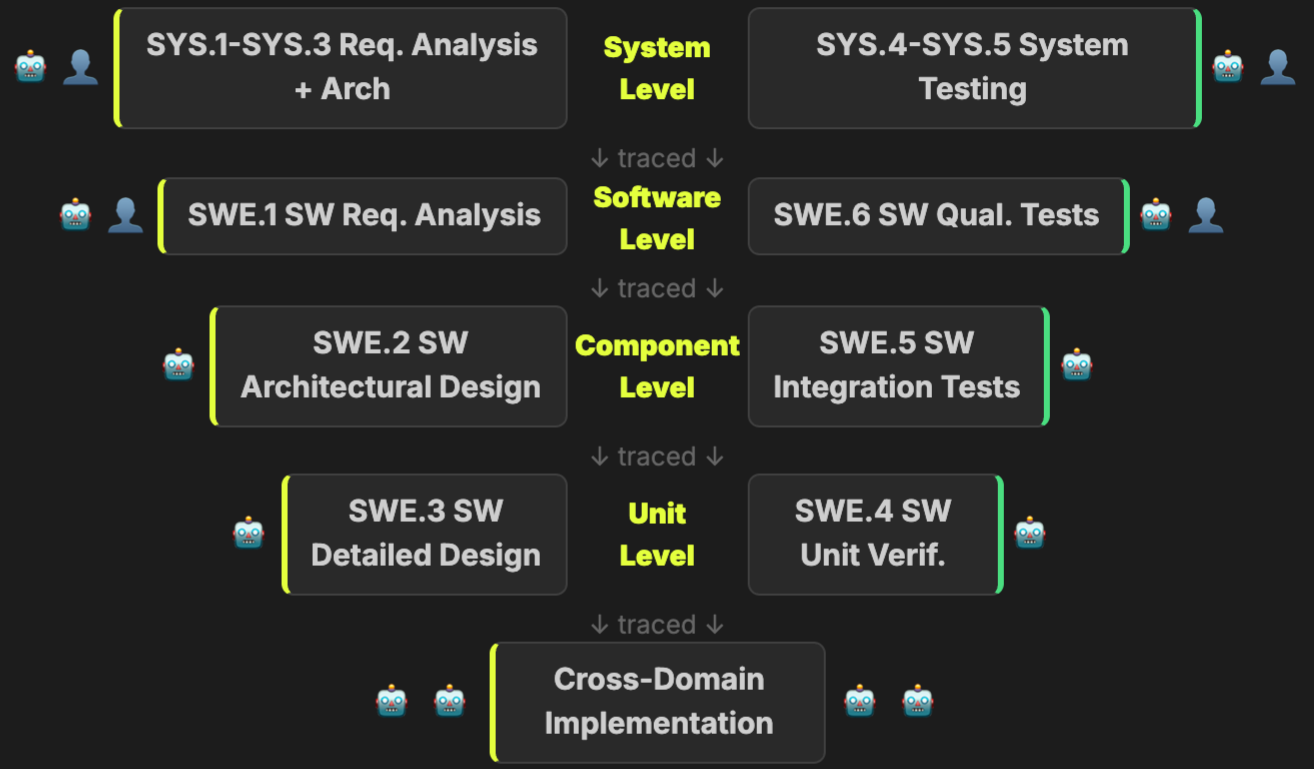
One prompt. Full V-Model. Fully traced.

USE BLOCKS

"Add an auto-tinted sunroof feature — activate when cabin temp > 30 °C."
One intent triggers a cascade of traced, verified changes across the full V-Model.

OEM-SPECIFIC CURATED CONTEXT

- **Knowledge** — Public coding examples, general practice knowledge →
- **Standards** — A-SPICE, MBSA, SysML, MISRA, AAOS … →
- **Guides** — OEM process descriptions, coding guidelines, AUTOSAR guides … →
- **Context** — Existing repos, test reports, issue tracking, change requests … →



Multi-Agent Orchestrator — powered by GH + MSFT + UB

Sphinx-Needs manages linked objects in Git. Changes are traced end-to-end. **ubCode** provides real-time AI guardrails via Ontology & MCP context.

Slide concept credit: Frank Kaleck

Value Proof

One change. 15 artifacts. Fully traced.

USE BLOCKS

Erwin Roth from Audi tested this scenario on a real ISO 26262 brake system with **3 vehicle variants** :

> **The change:** Brake response time drops from **100 ms → 50 ms**. A single number changes at the system level - but the impact ripples through the entire Sphinx-Needs graph: requirements, specs, hardware, code, tests, and architecture across all variants.

V-MODEL LAYER	#	WHY IT MUST CHANGE
System Requirement	1	Root timing value — shared across all 3 variants
Variant Requirements	3	Each variant (PC · TR · MC) derives the new 50 ms bound
SW Specifications	3	Base spec + truck override + motorcycle override
HW Specification	1	Brake actuator timing constraint — shared
Implementations	2	PC + truck only (motorcycle inherits, no override)
Test Cases	2	PC + truck only (motorcycle inherits, no override)
Architecture Diagrams	2	Signal flow + timing diagrams via codelinks
Total	**15**	**Pharaoh found 100 % — ubCode validated every change before commit**

> **6 of 15 are hard:** they require Pharaoh to traverse the Sphinx-Needs variant graph or follow codelinks - the kind of cross-cutting dependencies that manual reviews routinely miss.

Case study credit: Erwin Roth, Audi

What Worked

Solving real problems — proven in real OEM projects

USE BLOCKS

🏭

Real programs

Deployed in production OEM projects

⚡

Value the core can't give

AI guardrails, real-time validation, full traceability under regulation.

💶

Now they pay

Not gratitude — budget. Because the outcome is indispensable.

PMF #2, in one line: **to get paid on top of open source, you must deliver value the open source cannot.**

Some Learnings

PMF for the business model = solving real problems

USE BLOCKS

Open source alone

Brings trust, reach and community — but not enough revenue by itself.

Value on top

Customers pay when the layer above the OSS removes real pain in their workflow.

PMF #2

The business model works when it consistently delivers measurable value.

So the question is not: what can we sell? It is: what value can we create that OSS cannot provide alone?

The Real Asset

Your strongest OSS asset isn't code — it's community and people

USE BLOCKS





Open-source roots. **5M+ downloads.** A User Group that grew from **18 to 72** — the trust that exists before the revenue does.

Audience Question

The same pattern, at industry scale: S-CORE?

USE BLOCKS

QUESTION 1

S-CORE

What is the PMF for OSS?

QUESTION 2

S-CORE+

What is the PMF for the business model on top?

Let's talk

USE BLOCKS

MP

Maximilian Pabinger

CEO & Co-Founder · useblocks

📧 max.pabinger@useblocks.com

📱 +43 664 451 88 48

⭐ The open core that started it all: **github.com/useblocks/sphinx-needs**

If open source should win in automotive — fund the teams building it.