

TestSprite Funnel Audit

I spent time on your homepage, blog, and social presence, and wanted to share some outside-in thoughts on your funnel.

There's no doubt you're solving a real problem.

As GenAI code becomes standard, testing that high-volume code is life or death for dev teams.

The opportunity I see is making that urgency visible on your homepage and positioning TestSprite as the safety layer developers didn't know they needed.

Here are the 3 highest-leverage fixes to consider:

1. Lead with the AI code generation risk to build urgency and get more signups from the homepage

Your homepage leads with:

10x your AI-native dev speed with our autonomous AI testing agent handling verification.

The problem is that most developers don't yet understand **why they need TestSprite**.

They see another *10x AI agent* but don't feel the urgency and danger of the situation. Speed isn't necessarily the problem for these teams.

Here's what's actually happening: developers are shipping untested AI-generated code to production at scale. They're all doing it fast regardless.

A single Claude prompt generates 500 lines. Most of it ships without verification. Bugs slip through and outages happen, leading to disaster.

This is your hook.

Instead of leading with "10x your speed," lead with the problem TestSprite solves.

"Your untested AI code is shipping to production every day. TestSprite catches it before your customers do."

Then show the contrast:

Without TestSprite: Developer uses Claude to generate a feature. Ships it. Production bug appears 48 hours later. Fire drill. Customers mad.

With TestSprite: Same developer, same prompt. TestSprite automatically generates test cases, runs them, catches the bug. Feature ships clean. Customers happy.

Your hero section should call out this shift explicitly:

"AI is writing code faster than ever. But untested AI code is the next critical risk. TestSprite is your automated safety layer at scale."

Then below that, show the speed benefit: "Reduce testing time by 80%. Ship features faster. Deploy with confidence."

Own that positioning and you own the conversation.

2. Replace thought leader quotes with actual customer results to generate trust with real devs

Your homepage features quotes from Karpathy, Cherny, and Ng about the importance of testing. These are credible voices, but they don't prove TestSprite works.

Developers already know they need better testing. What they need to see is:

"Real developers used TestSprite and got real results."

Replace the thought leader section with a customer proof stack. Pull 2-3 real customer results and lead with specific metrics:

"Cut testing time by 80%"

A developer at [Customer] was spending 40 hours per week writing test cases. With TestSprite, same coverage in 8 hours. They redeploy that time to ship new features.

"Catch bugs before production"

An engineering team deployed untested AI-generated code. TestSprite caught 12 edge cases in automated verification. Zero production bugs. Customers never knew.

"Deploy 4.67x faster"

A team went from 2-week test cycles to 3-day cycles (77% time reduction). Same code quality. Lightspeed iteration.

The format: metric + context + result. Make it specific and relatable to devs everywhere.

If you don't have public customer stories yet, show anonymized case studies or Discord success stories. You can still quote Karpathy, but real examples count for more early on.

3. Own the GenAI code safety discussion on social to drive organic traffic

You have hundreds of blog posts, but your social presence (Twitter, LinkedIn) isn't hitting on the urgent problem TestSprite solves.

Right now your social posts are mostly about events, announcements and feature updates.

That's all great, but you could do more to draw from the thought leadership of your blog and the front-line experience of today's dev teams.

The conversation you should be leading is: "How and why should you safely deploy AI-generated code at scale?"

This is the moment to capitalize, because every developer is asking this question. TestSprite deserves to be at the center.

Instead of generic testing tips, lead with the pain and fear of the status quo:

"Your team shipped 500 lines of Claude-generated code today. How much of it did you actually test?" (makes people uncomfortable. good)

"AI code generation is 10x faster. But if testing doesn't scale, you're building a ticking time bomb." (pain point/urgency)

"Untested AI code in production is the new technical debt. Here's how we prevent it." (positions TestSprite as solution)

"Your developers are using Claude/Copilot. Your testing process wasn't built for this speed. That's a problem." (names the gap everyone's afraid of)

Repurpose your best blog content (and new material) into short-form video clips, battlecards, and slides showing:

- Real engineers explaining the testing gap with AI code
- Before/after examples of what TestSprite catches
- Clips of industry leaders talking about validation (+ how TestSprite solves it)
- Case studies: "Here's what broke when we didn't test AI code. Here's what TestSprite caught immediately."

The vibe should be: **Red alert. Alarm bells. This is urgent.** TestSprite is the solution.

Looking forward to hearing your thoughts.

Xander