

# From Copy-Paste Chaos to 5-Second Publishing

How AI-Assisted 'Vibe Coding' Transformed a Manual Content Workflow into an Automated Publishing Ecosystem

**Author:** Simeon Herbert — Agile Second Opinion

**Date:** March 2026

**Tools:** Lovable · Claude · React · Tailwind CSS

# Executive Summary

What once took 25 minutes of manual copy-pasting, reformatting, and cross-checking across three platforms now takes under 5 seconds with a single click. This white paper documents the complete transformation of a content publishing workflow — from a fragile, manual process involving LinkedIn, Buffer, and Squarespace into a fully integrated, API-driven ecosystem built entirely through AI-assisted development.

The project demonstrates how a non-traditional developer — an experienced Agile coach with deep technical intuition — can leverage AI tools (Lovable and Claude) to architect and build production-grade software through what has become known as **'vibe coding'**: directing AI with precise problem statements, iterating through feedback loops, and making architectural decisions that the AI alone would not have made.

## The Problem

Simeon Herbert writes a regular blog for his consultancy, Agile Second Opinion. The content serves dual purposes: LinkedIn thought leadership (via Buffer scheduling) and a permanent archive on the company website. The original workflow was:

- Write content in Claude or a text editor
- Copy into Buffer, manually format for LinkedIn's 3,000-character limit
- Copy the same content into Squarespace's CMS
- Manually fix line breaks, paragraph spacing, and markdown rendering
- Cross-check both versions for consistency
- Repeat weekly — with Squarespace changing its formatting behaviour regularly

The process took approximately 25 minutes per post and was error-prone. Squarespace frequently changed how it handled pasted content, meaning the formatting workflow broke almost weekly. For a busy consultant, this was an unacceptable time drain.

Metric	Before	After
Time per post	~25 minutes	~5 seconds
Platforms to manage	3 (LinkedIn, Buffer, Squarespace)	1 (Custom CMS)
Formatting errors	Weekly	Zero
Content API	None (web scraping via Firecrawl)	Native JSON API
Cost of content API	Firecrawl subscription	Free (built-in)
LinkedIn integration	Manual copy-paste	One-click Buffer publish

## The Solution Approach: Vibe Coding

Rather than hiring a developer or learning a framework from scratch, Simeon leveraged a methodology he calls **vibe coding** — using AI tools as collaborative coding partners, directed by clear problem statements and architectural thinking.

The toolchain consisted of **Lovable** (an AI-powered app builder that generates React applications) and **Claude** (Anthropic's AI assistant, used for code generation, debugging, and architectural discussion). Claude would write code that Simeon would review, refine, and deploy through Lovable.

Critically, the human role was not just 'prompting.' Simeon made key architectural decisions that shaped the entire system:

- **Replacing Squarespace entirely** rather than building a Squarespace integration
- **Building a public API** to replace the Firecrawl web-scraping dependency
- **Integrating Buffer's GraphQL API** for direct LinkedIn scheduling
- **Designing the LinkedIn text workflow** to handle posts that exceed the 3,000-character limit
- **Insisting on duplicate prevention** in the Buffer integration after spotting the risk
- **Directing the paste handler** to preserve formatting from Claude's output

## What Was Built

The final system is a complete content management and publishing platform, built with React 18, Tailwind CSS, and a cloud backend (database, authentication, serverless functions, and file storage). It replaces three separate tools with a single, cohesive interface.

### Public Website

A fully responsive site featuring a blog with Markdown rendering, SEO metadata, dynamic sitemap generation, a comment system, and pages for About, Contact, Testimonials, and FAQs.

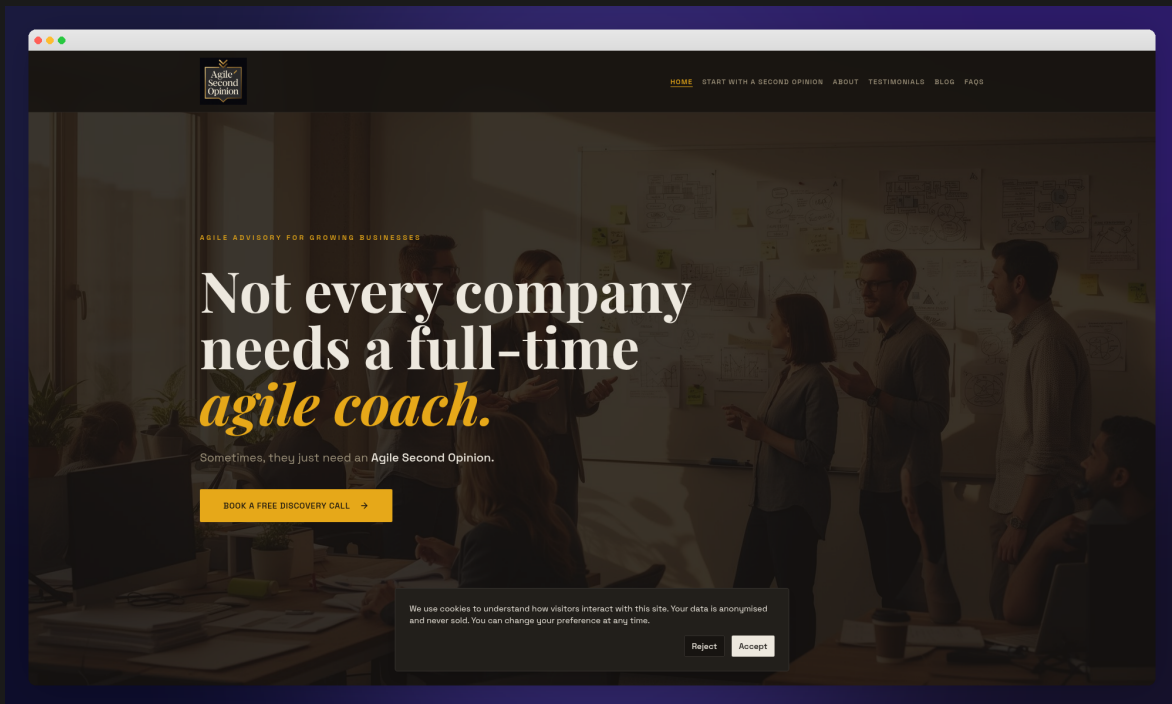


Figure 1: The Agile Second Opinion homepage

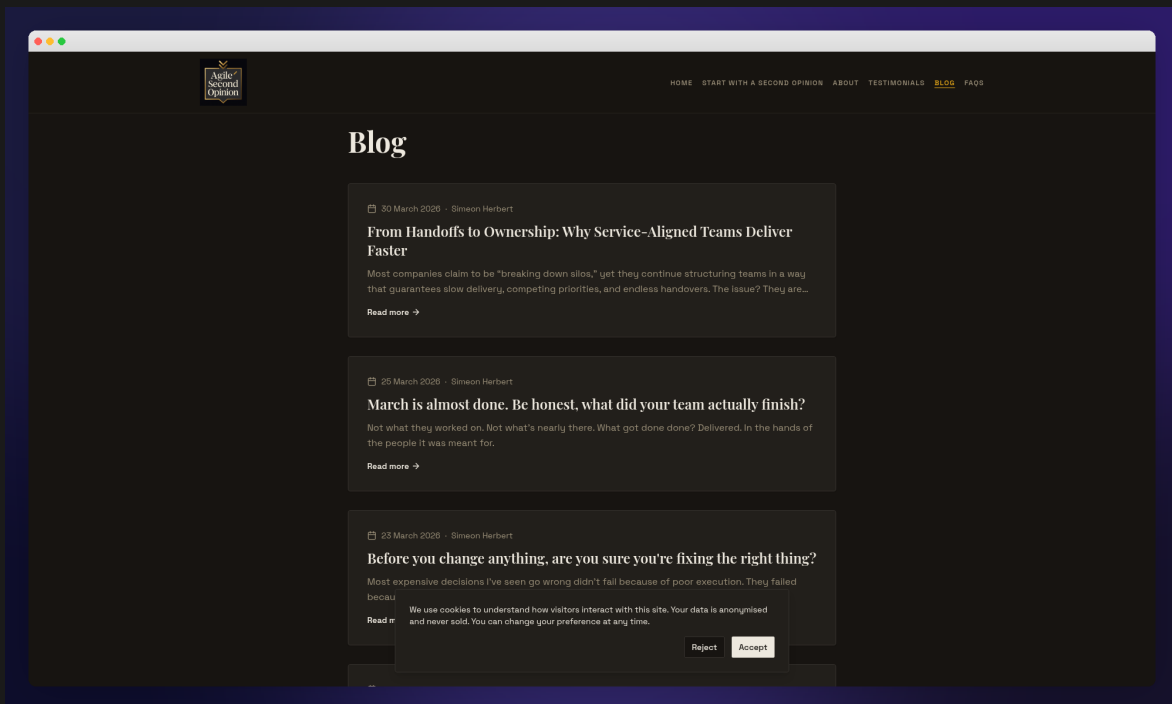


Figure 2: Blog listing with comment counts and responsive layout

## Admin Dashboard — Content Management

An authenticated admin interface provides a complete blog management system. The dashboard shows all posts with their status (Draft, Scheduled, Published), sorted chronologically, with quick-access API reference links and comment moderation.

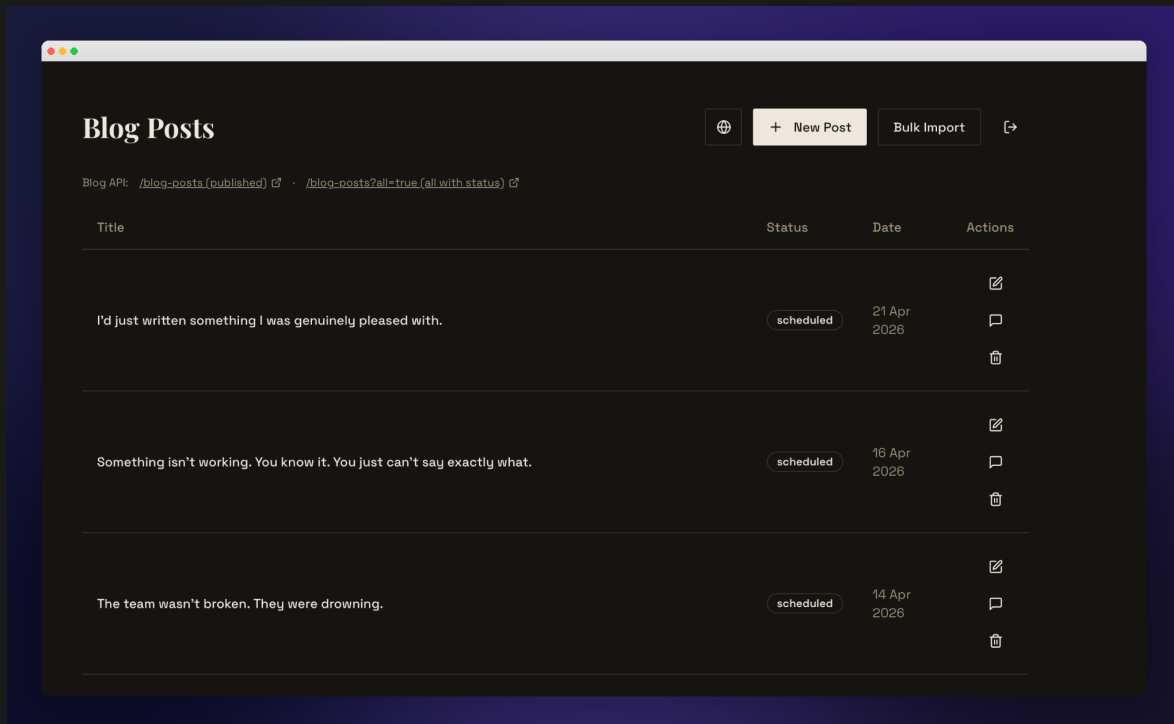


Figure 3: Admin blog list with status indicators, API links, and comment access

## Blog Editor — The Heart of the System

The editor is where the real workflow magic happens. It features:

- **Markdown editing** with a custom paste handler that preserves paragraph breaks from external sources like Claude
- **Auto-generated SEO slugs** from post titles, with manual override capability
- **Image upload** for feature images stored in cloud storage
- **Status management** — Draft, Scheduled, or Published with date picker
- **LinkedIn character counter** — a live display showing characters used out of the 3,000 LinkedIn limit

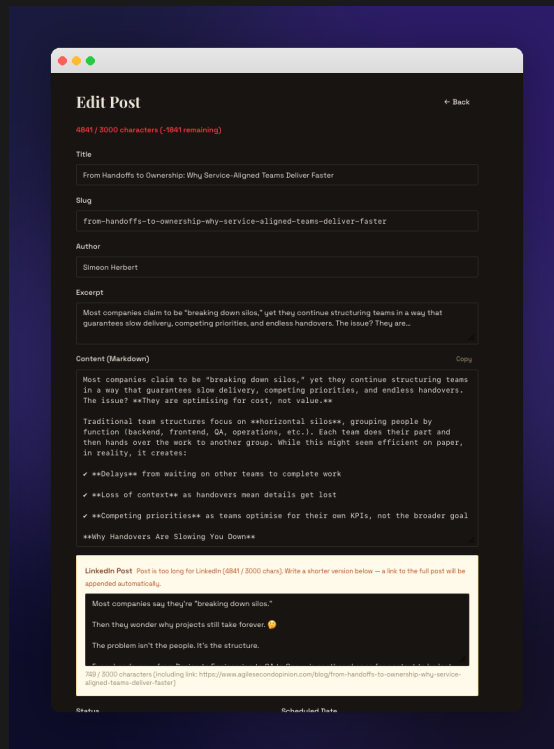


Figure 4: Blog editor with character counter, status controls, and Buffer integration

# Smart LinkedIn Publishing Workflow

One of the most elegant features is how the system handles the tension between long-form blog content and LinkedIn's 3,000-character limit. This was a key architectural decision driven by Simeon's understanding of his own content workflow:

## When Your Blog Fits LinkedIn

If the blog post is under 3,000 characters, the system uses the full content directly. The title is automatically prepended, and the post can be sent to Buffer with a single click — either **'Publish Now'** (immediate) or **'Schedule via Buffer'** (queued to the next available slot).

## When Your Blog Exceeds LinkedIn's Limit

This is where the system truly shines. When a post exceeds the character limit, the editor reveals a dedicated **'LinkedIn Text'** field. This allows Simeon to write a shorter introduction or teaser specifically for LinkedIn, while the full article lives on his blog. The character counter updates in real-time, showing exactly how many characters remain.

This means:

- The **full blog post** is always published on agilesecondopinion.com — no truncation, no compromise
- The **LinkedIn version** gets a purpose-written introduction that drives traffic to the full article
- The character counter provides **instant visual feedback** with warnings as you approach the limit
- No more guessing whether content will fit — the system tells you before you hit publish

## Duplicate Prevention

The Buffer integration tracks unique post IDs from Buffer's API. If a post has already been sent to Buffer, the system shows a clear warning and requires explicit confirmation before creating a duplicate — preventing accidental double-posts to LinkedIn. Because Buffer's API doesn't support updates (and deleting queued posts shifts the entire schedule), this safeguard was essential.

# The Content API: Replacing Web Scraping

A companion AI application previously used Firecrawl (a web-scraping service) to read Simeon's blog content. This was fragile, slow, and had a subscription cost. The solution was to build a native JSON API directly into the platform:

- **/blog-posts** — returns all published posts with full content
- **/blog-posts?all=true** — returns all posts including drafts and scheduled (admin use)
- The API is always up-to-date because it reads directly from the same database
- Any AI tool can now consume the content via a simple HTTP request — no scraping needed

# Technical Architecture

Layer	Technology	Purpose
Frontend	React 18 + Tailwind CSS	Responsive UI, Markdown rendering
Backend	Lovable Cloud (Supabase)	Database, auth, file storage
Serverless	Edge Functions (Deno)	Buffer API, content API, scheduled publishing
Social	Buffer GraphQL API	LinkedIn scheduling and tracking
Automation	Database cron job (5-min)	Automatic scheduled post publishing
SEO	Dynamic sitemap + meta tags	Search engine optimisation

## The Human in the Loop

This project is not a story of AI replacing a developer. It is a story of a skilled professional leveraging AI as a force multiplier. Key human contributions included:

- **Problem framing** — defining what needed to be built and why, not just how
- **Architectural decisions** — choosing to replace Squarespace entirely, building the API, designing the LinkedIn text workflow
- **Quality assurance** — catching bugs the AI missed, testing edge cases, verifying Buffer integration behaviour
- **Iterative refinement** — each feature went through multiple rounds of 'that's not quite right, try this instead'
- **Domain expertise** — understanding content workflows, LinkedIn's constraints, and the specific pain points to solve
- **Risk identification** — spotting the duplicate post risk before it became a production problem

## Results & Impact

The transformation delivered measurable improvements across every dimension:

- **Time savings:** 25 minutes → 5 seconds per post (99.7% reduction)
- **Error elimination:** Zero formatting errors since launch
- **Cost savings:** Eliminated Firecrawl subscription
- **Content sync:** Blog and LinkedIn always consistent, with smart handling of length differences
- **API ecosystem:** Content accessible to any AI tool via JSON API
- **Extensibility:** New features (comments, SEO, scheduling) added in hours, not weeks

This project demonstrates that the future of software development isn't just about writing code — it's about knowing what to build, how to direct AI tools effectively, and when to step in with human judgement. Simeon's background in Agile coaching — understanding iterative delivery, feedback loops, and stakeholder communication — proved to be the perfect foundation for vibe coding.

— **Simeon Herbert, Agile Second Opinion**

[agilesecondopinion.lovable.app](https://agilesecondopinion.lovable.app)