We Redesigned for Agents. It Made Everything Better for Humans.

Agent Experience Is Human Experience

How AX Is Rewriting the Software Development Lifecycle

Dana Lawson • CTO Netlify

A New Era of Creators

The Builder Is No Longer Just a Developer

The builder persona has expanded. Developers are still builders — but now so is everyone else.

500M+ apps by 2028

IDC predicts more apps will be built in the next few years than the past 40 combined.

Agents are the new IDE

Claude, Cursor, v0, Bolt — non-engineers are building production apps through conversation.

The global builder

People worldwide — therapists, teachers, small business owners — are becoming software creators.

Meet Torria

Healer. Drummer. Dreamer.

My best friend Torria is a master massage therapist, a vibrant drummer, and an absolute force of nature.

Her vision? An all-inclusive wellness retreat in Barbados. She doesn't write code. She writes intentions.

Massage Therapist | Drummer | Wellness Guide | Dreamer | Builder

torrialeelmt.com

@torrialeelmt

The Wall

Stuck on "Git"

She prompted an agent to build her a website.
But she hit a wall — over and over again.

```
git commit -m "initial commit"

fatal: not a git repository
```

"She had no idea what the agent was telling her."

The tools assumed she was an engineer. The agent spoke in a language she didn't understand. This is the gap AX must close.

A Paradigm Shift

"You don't need to worry about git."

Telling her not to worry about git is a paradigm shift. We are removing the guardrails of the past to let new builders create.

Why AX Matters

AX makes it possible for agents to **just work** — abstracting away complexity so anyone with intent can build.

At Netlify, this means agents handle git, CI/CD, DNS, and deployment. The builder focuses on their vision. The platform handles the rest.

The Boundaries are Collapsing

A new discipline is emerging: Agent Experience (AX) — designing systems where humans and AI agents collaborate seamlessly to build, test, and deploy software.

The Landscape

The Rise of Agentic Development

2024–2026: A Cambrian explosion of AI-powered development tools has emerged — and agents are becoming first-class participants in the SDLC.

Claude Code
Agentic CLI coding

Cursor / Windsurf
AI-native IDEs

v0 / Bolt / Lovable
Prompt-to-app builders

GitHub Copilot Agent
Autonomous PRs & issues

These tools share a common pattern: they treat **intent as the input** and **working software as the output**. The infrastructure beneath must evolve to support this.

What's Changing

From Specs to Intent

AI-native workflows begin with intent — not PRDs, not Jira tickets, not spec docs.

The Input

- Natural language prompts
- Problem descriptions
- Sketches, screenshots, references
- Workflow definitions

The Output

Code moves immediately into real engineering workflows: preview deploys, automated tests, and production pipelines.

At Netlify, our Agent Runners turn prompts into deployed preview URLs in minutes — not days.

What's Changing

From Sequential to Shared

HISTORICAL HANDOFF

PM › Design › Engineering › QA › Deploy

Weeks of handoffs. Context lost at every boundary.

SHARED CREATION PIPELINE

- Product teams generate working prototypes with agents
- Designers refine experience directly in code
- Engineers validate architecture and set guardrails
- Agents test, deploy, and monitor autonomously

What's Changing

Autonomous Development Loops

CI/CD evolves from a passive pipeline into a continuous, agent-driven feedback loop.

Netlify's deploy preview and build infrastructure already provides the observability agents need. When a build fails, an agent can read the logs, diagnose the issue, and submit a fix — without a human in the loop.

AX in Practice

How Netlify Is Building for Agents

We're not just talking about AX — we're shipping it. Here's what agent-native infrastructure looks like at Netlify.

Agent Runners

Agents run in sandboxed environments with full access to the Netlify platform — deploy, configure, and iterate without human intervention.

Deploy Previews for Agents

Every agent-generated change gets a unique preview URL. Humans review the output, not the process.

AI Gateway

Multi-provider AI inference at the edge — no API keys to manage, no vendor lock-in. Agents pick the right model for the task.

Platform Primitives

Blobs, Edge Functions, Forms, Identity — composable building blocks agents can assemble without understanding infrastructure.

The Problem

Fragmented Systems

Most organizations are introducing agents into systems designed for human operators.

The mismatch is creating real friction.

Fragmented Context

Agents can't see across service boundaries

Inconsistent Interfaces

Every API speaks a different dialect

Hidden Workflows

Tribal knowledge locked in human heads

Limited Observability

Agents can't inspect what they can't see

Architectural Shift 1

From APIs to Capabilities

Traditional systems expose endpoints. Agent-native systems expose intent-level operations that agents can reason about.

>_ Traditional Endpoints HTTP

```
POST /api/v1/sites
PUT /api/v1/sites/{id}/build-settings
POST /api/v1/sites/{id}/deploys
GET /api/v1/sites/{id}/ssl
PATCH /api/v1/dns_zones/{id}
```

Agent Capabilities

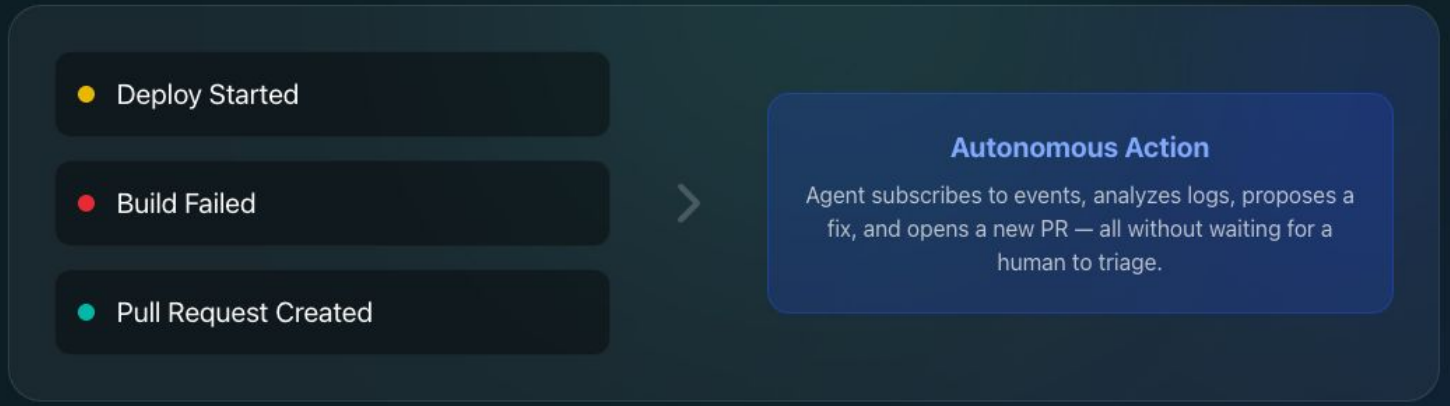
```
→ create_a_site()
→ deploy_repository()
→ attach_domains()
→ provision_edge_compute()
```

Agents reason about what they want to accomplish, not which endpoints to call in what order.

Architectural Shift 2

Request-Response to Event-Driven

Agents perform best when they can observe system behavior and act autonomously — not poll and wait.



Architectural Shift 3

Making Architecture Legible

The tribal knowledge that humans carry in their heads — service relationships, deployment quirks, failure modes — is invisible to agents.

Blueprints

At Netlify, we created Blueprints — a shared architectural knowledge base that describes how services relate, cross-service contracts, deployment dependencies, and request flows. Agents read these before making changes.

> *"The goal isn't better documentation — it's ensuring AI agents understand the system they're operating inside before making changes."*

A Continuous Loop

Agent-Orchestrated Systems

From static toolchains to continuous human-agent collaboration loops.

1. PM describes feature in natural language
2. Agent generates working prototype
3. Agent writes and runs tests
4. Deployment agent provisions preview
5. Observability agents monitor health
6. Optimization agents suggest improvements

The human stays in the loop — but the loop moves at machine speed.

The Responsibility

Trust & Safety in Agent Systems

When agents can deploy code, provision infrastructure, and modify production — trust isn't optional. It's the foundation.

🔒

Sandboxed Execution

Agents run in isolated environments. They can't escape their sandbox or access resources they're not explicitly granted.

👁

Human-in-the-Loop

Deploy previews let humans review agent output before it goes live. The agent builds; the human approves.

📋

Audit & Rollback

Every agent action is logged. Every deploy can be rolled back instantly. Trust is built on transparency.

The Organizational Impact

Lowering the Barrier

When agents lower the barrier to building, everyone participates — and the role of engineering transforms.

Product

Generate working prototypes from prompts. Test hypotheses in hours, not sprints.

Marketing

Build interactive campaigns and landing pages. Ship experiments without engineering tickets.

Support

Create internal tools and dashboards. Solve customer problems with custom solutions.

> Engineering evolves from implementing every feature to designing the systems that make safe creation possible.

Summary

Key Takeaways

- The builder persona has expanded — agents serve everyone, not just engineers.
- Legacy systems must evolve from APIs to capabilities agents can reason about.
- Trust and safety are foundational — sandboxing, human-in-the-loop, full audit trails.
- Start with structured errors, event-driven signals, and legible architecture — these are AX practices you can ship today.
- Designing for agents makes everything better for humans too. AX is human experience.

The Evolution of Value

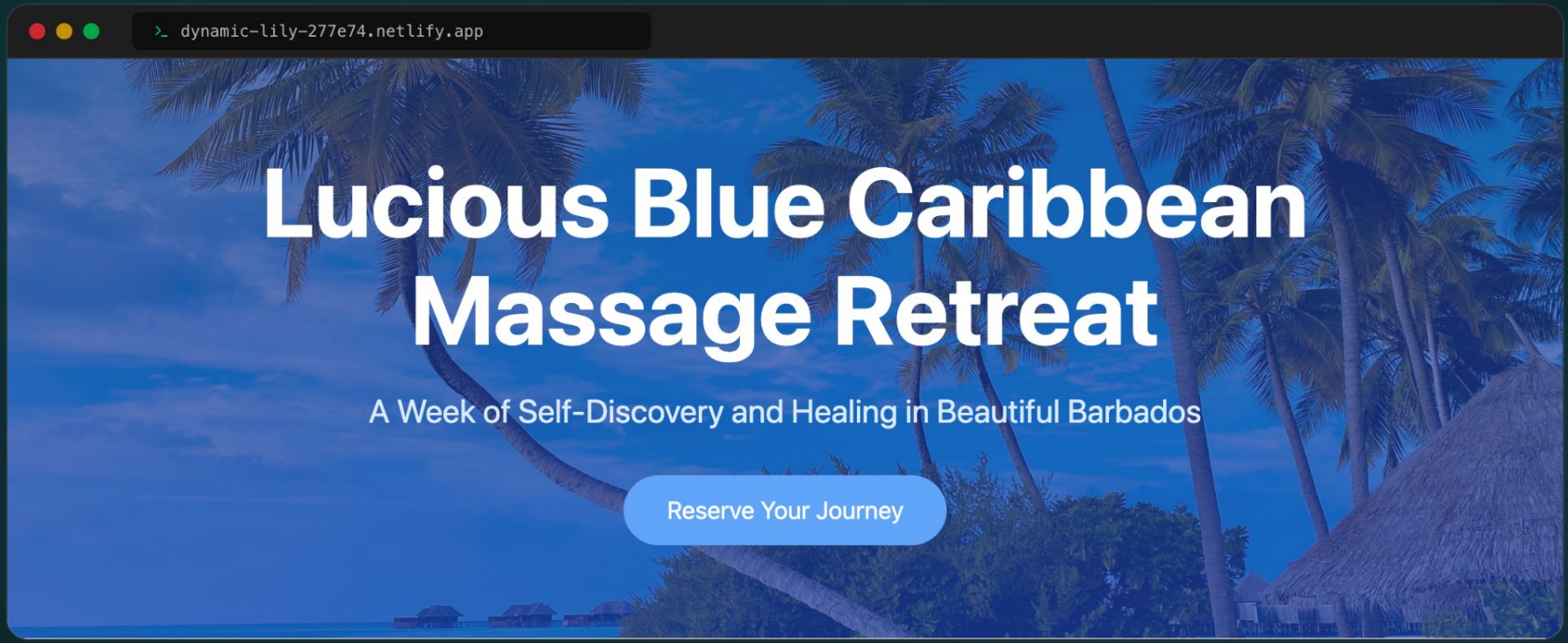
The New Scarce Resource

> "When anyone can generate software, code is no longer the scarce resource."
>
> — Matt Biilmann, CEO Netlify

Taste. • **Judgment.** • **Architecture.**

She Is A Developer

I didn't build this. She did.



Thank You

Agent Experience Is Human Experience

>_ netlify.com/ax