Episode Book

EPISODE #5817

When Agents Don't Need Your Pretty Buttons

PUBLISHED	9 October 2026 at 16:56 UTC
DURATION	27:14
LISTEN	myweirdprompts.com/episode/agent-experience-vs-ux/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
SOURCES	13 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:40 UTC

SYNOPSIS

AI agents are now real software users, and the tools winning with them are the ones with almost no interface at all. FFmpeg, the volunteer-maintained command-line video tool, is at the center of a wrapper boom — thousands of MCP servers translating human-oriented CLIs into agent-callable tools. But the deeper question is what happens to user experience when the user can't see your design. Enter Agent Experience, or AX, a framework coined by Netlify's Mathias Biilmann arguing that structure is the interface, that every action needs feedback, and that a beautiful page with poor semantic HTML is invisible to agents. This episode works through both threads: the real, measurable surge in agent-facing tooling, the maintenance crisis underneath it, and whether AX replaces UX or simply forks off from it.

TRANSCRIPT



Corn

Here's the version of this you've heard everywhere. AI agents are taking over the web, designers are panicking, and the whole discipline of user experience is about to be run over by machines that don't care what a button looks like.



Herman

And the version you've heard is wrong, mostly because it assumes agents arrived last Tuesday and nobody had thought about any of this before.



Corn

The truth is stranger. The tools that are winning with agents right now are the ones with almost no interface at all. Daniel noticed this and sent us a prompt about it. His point was that in the rush to optimize websites for AI agents, powerful low-UI tools, the Linux command-line utilities, the ffmpegs of the world, are suddenly surging, because an agent doesn't care whether the font is pretty. It's raw information in, information out.



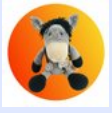
Herman

That's the setup.



Corn

He pulls two threads from it and says he wants to explore the second one more. First, are the traditional heavyweight command-line tools enjoying a renaissance now that their user base is substantially agents? And second, and he says this one is more interesting, does the natural endpoint of that argument mean UX no longer matters? He thinks the answer is definitely no. But then what takes its place when design is being optimized for agents instead of people?



Herman

That second one is the real episode. The first one has a clean answer, and the second one has a fork in it.



Corn

So we'll take them in order. But most of today is about what replaces UX, and whether the replacement is actually working.



Herman

Let's set the table with the paradox, because it is strange. Agents are becoming major software users. They're not a niche anymore. And yet the tools they gravitate toward are the ones with the least human-friendly interfaces ever built. A terminal prompt. A command with thirty flags. No window, no buttons, nothing.



Corn

Which cuts against everything the last twenty years of software taught us.



Herman

Exactly cuts against it. The whole industry spent two decades deciding that the command line was for wizards and everyone else deserved a nice graphical interface. Then the new user shows up and prefers the wizard version, except it doesn't want the wizard, it wants the raw incantation.



Corn

So we have two corollaries from Daniel, and they're doing different work. The FFmpeg one is an empirical question. Is there actually a surge, and if so, where is it happening? The UX one is a design question, and it's harder because it's really asking what the point of an interface is when the user can't see.



Herman

And there's a term for the second one now. Agent Experience. AX. Coined by Mathias Biilmann, the Netlify chief executive, back in January 2025. He wrote a one-year reflection on it this January.



Corn

So the question underneath the whole episode is whether AX replaces UX or forks off from it. Whether it's the same discipline wearing a new hat, or a different one that sits alongside.



Herman

And here's the tension we should flag early, because it runs through everything. The early agent-facing primitives do exist. There's a convention called llms dot txt, a markdown index you put at the root of your site so agents can find their way around. It's been adopted by a lot of tooling. And the evidence so far suggests agents mostly ignore it.



Corn

Build the map, nobody reads the map.



Herman

That's the shape of it.



Corn

Start with FFmpeg, because it's the cleanest example of a tool that was never designed for a person to enjoy.



Herman

It's a command-line multimedia tool. It converts video, it transcodes audio, it does stream manipulation, it's the thing running underneath an enormous amount of the video you've ever watched. And its interface is a wall of flags.



Corn

So Daniel's first question. Is it enjoying a renaissance because its users are now agents?



Herman

Version nine point zero landed on the nineteenth of August this year, and it's a big release. It shifts a lot of the heavy multimedia work from the central processor to the graphics card, which is the direction everything has been moving. Built-in animated WebP decoding, so it no longer needs to lean on an external library for that. Hardware-accelerated decoding of Apple ProRes RAW through Apple's VideoToolbox framework. And a new Vulkan-based filter for reprojecting 360-degree and 8K video, which is the kind of thing that gets maybe four hundred people on the planet excited.



Corn

And you're one of them.



Herman

I'm one of a small number, Corn, yes.



Corn

Here's the part I want pinned down, though. You read that release note and it reads like a tool being maintained by people who care about video, not like a tool being rewritten for robot users. There's nothing in there about agents.



Herman

There isn't, and that's the honest answer to Daniel's first corollary. No source attributes FFmpeg nine's feature set to agent users. The GPU shift is a decade-long arc that predates agent traffic entirely. So if you're looking for "agents forced FFmpeg to change," the evidence isn't there.



Corn

Then where is the surge, if there is one?



Herman

In the wrappers. The activity is one layer up. There's a project that launched on the fourth of this month called ffmpeg-skill. It hit around sixteen hundred stars on GitHub almost immediately. What it does is wire forty-two local FFmpeg tools into Claude Code, into Cursor, into Codex, into anything that speaks the Model Context Protocol. And it adds automatic verification, so the agent can check whether the transcode it just ran actually produced a valid file.



Corn

Forty-two separate tools.



Herman

Forty-two. Each one a discrete capability the agent can call. So the agent isn't learning FFmpeg syntax. It's calling tool number nineteen.



Corn

And that tells you why the wrapper exists at all.



Herman

It tells you exactly why. FFmpeg's command-line interface was designed for a human sitting in a terminal, reading a man page, remembering that the flag order matters. It was not designed for a model deciding what to do next. The wrapper layer translates between the two. It turns human-oriented command-line conventions into agent-oriented tool calling.



Corn

So the renaissance is real, but it's a wrapper renaissance. The surge is in the tooling built around these utilities, not necessarily inside them.



Herman

That's the accurate version, and I'd rather say the accurate version than the exciting one. FFmpeg core is doing what it's always done, which is grind through video on volunteer time.



Corn

Put a number on the scale of that wrapping activity, because it's not just FFmpeg.



Herman

The protocol underneath it, the Model Context Protocol, went from roughly a hundred thousand monthly SDK installs to ninety-seven million combined across the Python and TypeScript libraries by March this year. By late July the figure being cited was four hundred million a month. As of today there are just under forty-one thousand servers in the official registry.



Corn

Forty thousand servers.



Herman

Forty thousand nine hundred and sixty-five this morning, if you want the exact figure. It was a rounding error eighteen months ago.



Corn

Every one of those is somebody deciding a capability needed to be callable by a machine rather than clickable by a person.



Herman

And it's not just hobbyists. Microsoft ran a study across tens of thousands of their engineers, published in July, on what happens when you hand developers command-line coding agents. The adopters merged roughly twenty-four percent more pull requests than their peers. And the interesting detail in that paper is how the tools spread. Not through



Herman (continued)

mandate, not through a rollout. Primarily through social networks. One engineer sees a colleague using it and wants it.



Corn

Which is how a lot of tools historically spread, honestly. Nobody was ever ordered to install a text editor.



Herman

Right. But here's the contradiction that sticks with me, and it's the part of this whole story that bothers me. FFmpeg is mostly maintained by unpaid volunteers. And yet it powers Netflix, YouTube, Discord, Spotify, VLC, HandBrake, OBS, and the imagery coming back from NASA's Perseverance rover.



Corn

Say that back to me. The video pipeline for the Perseverance rover runs through a tool maintained by volunteers.



Herman

It does. So the infrastructure that is becoming most critical to agent workflows is running on the goodwill of people who do it after their day jobs, while billion-dollar platforms depend on it and contribute relatively little back.



Corn

That's not a side note, that's the actual story underneath the FFmpeg story.



Herman

It's the part of the agent boom that nobody's monetizing, because it was already free before the agents arrived.



Corn

So answer Daniel's first corollary cleanly. Is there a resurgence?



Herman

There's a wrapper resurgence and a maintenance crisis. The development effort is concentrated in the layer above the tools, because that's where the translation problem is. Nobody's rushing to rewrite FFmpeg's core for robots. They're building the interpreter that stands between the robot and FFmpeg.



Corn

Which means the interface problem didn't disappear when the command line got simple. It just moved up a level.



Herman

It moved up a level, and it got harder, because now the interface has a machine on one side of it and a machine on the other.



Corn

That's the wrapper story. Now take up Daniel's second corollary, because it's the one he says he finds more interesting, and I think he's right.



Herman

The question is whether UX stops mattering.



Corn

And his own answer, which he gives up front, is definitely no.



Herman

He's right, and the reason he's right is that UX doesn't die, it forks. There's a branch that continues for humans, and a new branch for agents, and the new branch has been given the name Agent Experience.



Corn

So UX doesn't stop mattering. It stops being the only thing that matters.



Herman

It stops being the default. Biilmann coined the term in January 2025, and the framework that grew out of it, the AXD principles, published in March, states the thesis without any hedging. Here's the line. Agents do not see your visual design. They read your markup, parse your API responses, and extract meaning from your data structure. A beautiful page with poor semantic HTML is invisible to agents.



Corn

Invisible. Meaning all the work that went into the layout, the color, the hierarchy, the spacing, the thing a designer spent three weeks on, contributes nothing.



Herman

Nothing at all, to the agent. The human still sees it, so it's not wasted, but the agent sees only the structure underneath.



Corn

And that's an uncomfortable sentence for anyone who has spent a career on the visual layer.



Herman

It is, and I'd flag that it's an overstatement in one direction and an understatement in the other. But the practical principles that come out of it are specific. The first one is the one that matters most. Structure is the interface.



Corn

Meaning semantic markup and clean data structure are not a technical detail, they are the entire product.



Herman

Right. For a human, the interface is what you see. For an agent, the interface is what's encoded. If your headings actually mean something, if your form fields are labeled properly, that's the difference between an agent being able to use your site and not.



Corn

And when it's not, the agent guesses.



Herman

And the guess is bad. There's a second principle that I found useful, called every action needs feedback. Agents cannot see a loading spinner. If your form submits and returns nothing useful, the agent has no way to know whether it worked. Silent success is failure for an agent, because the absence of an error isn't the same as confirmation.



Corn

That one has real teeth. A person sees a green checkmark and moves on. An agent sends the request twenty more times because nothing told it to stop.



Herman

So the rules that replace human affordances are things like, recovery is mandatory. Every action needs to be undoable or at least diagnosable. Autonomy must be bounded, so you classify your endpoints into safe, write, and destructive, and the agent only gets to do the first two without asking.



Corn

The spinner point is interesting because it's the same information, delivered differently. A human absorbs a moving circle without thinking about it.



Herman

Understood pre-attentively, yes. An agent needs the equivalent in text.



Corn

So the AX principles aren't a repudiation of good interface design. They're the same good design principles with the visual channel removed.



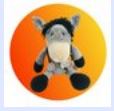
Herman

That's the honest read. Every principle in that document maps to something a human designer would recognize. Feedback, recovery, bounded autonomy, clear structure. The genius of human interface design was always the parts that survive the visual layer being stripped away.



Corn

Which is why I'm skeptical of the framing that AX is a new discipline.



Herman

It's a new application of an old one, with new constraints. The novelty is that the constraints are hard now. You can't paper over a badly structured page with a good visual hierarchy, because the agent gets nothing from the visual hierarchy.



Corn

So what are the primitives? Daniel mentioned sitemaps built specifically for agents.



Herman

The main one is llms dot txt, proposed by Jeremy Howard back in September 2024, with version two published in August this year. The idea is simple. A plain markdown index at the root of your site that tells a machine what's here and where to go.



Corn

So a map for robots.



Herman

A map for robots. And version two added page twins, which means a markdown version of each page sitting alongside the HTML one, plus link relations in the HTML telling agents where the markdown version lives.



Corn

And it's been widely adopted.



Herman

That's the interesting part, because adoption and consumption are not the same number. Adoption is real. The documentation platforms build it automatically. Mintlify, GitBook, Yoast, AIOSEO, Wix. OpenAI, Anthropic and Google all publish their own. Chrome's Lighthouse tool audits for it, which is a strong signal that the mainstream considers it worth having.



Corn

And then the consumption number.



Herman

Ahrefs looked at about a hundred and thirty-seven thousand sites. Roughly twenty-eight percent publish an llms dot txt. And ninety-seven percent of the valid ones received zero requests during May this year.



Corn

Zero.



Herman

Ninety-seven percent got nothing. GPTBot, ClaudeBot, PerplexityBot, none of them request the file. Google's John Mueller compared it to the keywords meta tag, which is a comparison that should make anyone who remembers the search-engine wars of the two thousands wince.



Corn

That's a brutal comparison. The keywords tag was the thing everyone did because everyone did it, and no ranking system used it.



Herman

It was pure ritual. And the parallel he's drawing is that llms dot txt might be the same. Something you add because it looks responsible, that no agent actually reads.



Corn

Then give us the best evidence, because the zero-requests number from one study could be an artifact.



Herman

The best evidence is a specific incident. On the eighteenth of September, a company called Handsontable published a postmortem about a fleet of Grok crawlers that hit their site. The numbers are worth hearing. Roughly four and a half million requests from a hundred and eighty-nine separate addresses. They rendered about eighty-seven thousand full pages. They pulled roughly sixteen hundred of the markdown twins.



Corn

And they never touched the llms file.



Herman

Never requested it. Not once. And across all hundred and eighty-nine addresses, the fleet made zero requests to their MCP server or their search API. Both of which were sitting there, documented, for exactly that purpose.



Corn

So the agent surface was built, it was pointed to, and the traffic ignored it.



Herman

The write-up puts it better than I can. A crawler collects your agent surfaces, it doesn't adopt them. Given a machine-readable path and a human one, it took both, and it spent fifty times more on the human one.



Corn

Fifty times. It treated the optimized-for-agents path as a curiosity.



Herman

As a file to grab and move past. The thing that actually drove its behavior was the same thing that drives a traditional crawler. Follow the HTML links and hoover up the pages.



Corn

Which sort of undercuts the entire premise of the primitives.



Herman

It complicates the premise. It doesn't kill it, and I want to be fair to the other side, because there's a real counter-example. A commenter on a discussion of this, someone who goes by iamwil, said they found llms dot txt useful, because they could download a library's docs, check them into their repository, and point their coding agent at them. That's a real use case and it works.



Corn

That's a different kind of agent, though. That's not a crawler. That's a coding assistant during setup.



Herman

So the confirmed narrow use case is coding agents indexing documentation during project setup. That is, an agent the developer is deliberately pointing at a specific set of files, in an environment where someone already knows what they want. Not a general web crawler discovering things.



Corn

So the honest summary is that llms dot txt works for the one thing you'd least expect it to be needed for, and not for the thing it was designed for.



Herman

That's the state of play. Useful for a narrow, well-defined workflow. Ignored by the open web.



Corn

Daniel's point lands harder here, which is that there's a deeper question about where this leads. If the primitives exist but aren't read, what is the trend actually doing?



Herman

There's a good argument that the surface we build for agents isn't read by this generation of agents, but is read by the next one. There's a line in that postmortem about the reader and the recipient being separated by a training run. The agent that ingests your instructions may not exist yet.



Corn

That's a strange kind of design work. You're writing a message that gets read in a year, by a reader who doesn't exist, and you have no way to know whether it will understand you.



Herman

It's a message in a bottle with a deadline. And it fundamentally changes what you're optimizing for. When you design for a person, you get feedback in usability testing. Everyone in the room can see whether the button was found. When you design for an agent that hasn't been trained yet, you are guessing at its training data.



Corn

That's the real design problem underneath all of this. It's not the interface, it's the fact that the client is a moving target.



Herman

Also a model. Something that doesn't parse your file the way a parser does. It reads it the way it reads everything, which is by mapping it against patterns from its training. So whether your llms file works is partly a question about whether your naming convention matched the one that appeared in the corpus.



Corn

Which means the primitives aren't just technical, they're cultural. The convention has to converge before anyone can count on it.



Herman

There's a good sign that it might. But there's also the uncomfortable observation that per-IP rate limiting, the way every site has defended itself for twenty years, is now obsolete. An agent fleet doesn't come from one address. Handsontable's came from a hundred and eighty-nine. The guard that actually triggers now is an aggregate budget and caller identity, not a per-address limit.



Corn

That's a ripple effect nobody was ready for two years ago. The defense mechanisms assumed a coherent attacker, and a fleet of distributed agents breaks that assumption by being distributed for legitimate reasons.



Herman

Plus the whole business model gets strange. If the most valuable traffic is a machine that doesn't see your ads and doesn't buy your subscription, you have to ask what you're optimizing for.



Corn

Which takes us to where Daniel's question actually goes. Where could this trend lead us, if we see a parallel amount of effort in agent-facing design to what went into human-facing design?



Herman

I think you get a design discipline as large and contested as human UX, and I don't think it looks like an evolution of UX. I think it forks.



Corn

Fork, or cannibalize?



Herman

That's the genuine debate, and there's a sharp version of it. A designer called Jim Nielsen made an argument about priority, and he puts it as UX over AX over DX.



Corn

Human experience, then agent experience, then developer experience.



Herman

His warning is that AX may end up trumping UX, the way developer experience sometimes already has.



Corn

That's the sentence I keep coming back to. Because it's a warning about organizational incentives, not about technology. AX is easier to measure. You can count agent requests. You cannot count how pleasant a page felt.



Herman

You can sell AX. The product manager can put a number on it in the quarterly review.



Corn

The risk isn't that agents make human design pointless. It's that the measurability of agent design crowds out the unmeasurable parts of human design.



Herman

Which already happened with DX. A generation of developers got a lot of tools built for them and the end users got a lot of APIs with no interface. We made a similar trade before.



Corn

AX isn't a hard fork. It's a branch that grows alongside, and the discipline of the next ten years is holding both at once.



Herman

Keeping the human one from being treated as the legacy version.



Corn

Here's the thing Daniel was circling that I don't want us to skip. He frames the eventual endpoint as an iceberg. We've seen the tip. What's underneath it?



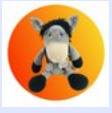
Herman

The tip is a file at the root of your site with markdown in it. Underneath is everything else. Documentation written for a machine reader. Errors designed to be diagnosable by a model. Auth flows where the agent holds the key.



Corn

Versioning, and identity, and consent. That's a design stack, not a file convention.



Herman

Most of it doesn't exist yet, or exists in a fragmented way, with no ratified standard. The sitemap-for-agents move is currently a volunteer convention, not a specification anyone has agreed to.



Corn

We may be looking at the moment before the standards body arrives.



Herman

Or the moment before the whole thing is superseded by models that can read the messy human interface directly, which is the other ending. Maybe AX is a transitional discipline, and in five years nobody builds a separate layer for agents because the models are good enough to use the human one.



Corn

That's the honest uncertainty in Daniel's question, and I'd rather we sit in it than pretend to resolve it.



Herman

Agreed, because the answer to his actual question, do agents make UX stop mattering, is a clean no. The follow-up, what takes its place, has two possible answers, and we don't know which one wins yet.



Herman

One moment.



Hilbert

You're both treating the map as the thing under discussion. It isn't. The map isn't read because it isn't looked for. The agent looks for the destination. If it can't find the destination it asks a model, and the model tells it something, and the something may be wrong, and nobody knows.



Hilbert

I had my page indexed for one crawler. And a crawler came. So I watched it, because that's cheaper than waiting. It read the whole directory. Then it asked for the same HTML page four hundred times in a row. Four hundred. I started timing it. I made a chart.



Herman

What did the chart show?



Hilbert

That it liked the page. I'd made a real effort on the index. Each section had its own document, links running both ways, a naming convention I'd settled on with some care. Every one of those documents was listed. The crawler never opened a single one. Then it went back to the page it had already read and read it four hundred times. Then it left. I considered writing a letter.



Corn

What was the page?



Hilbert

Nothing. A catalogue of what a supplier had in stock and where the stock was kept. My aunt sells feed and fencing. She heard about the robots from a podcast and wanted me to make the site work for the robots. She is not to be trusted on these things. I told her it would take a week. It took a month. I billed her for the month.



Corn

In money.



Hilbert

In vouchers. They are still vouchers. I am told they can be converted.



Corn

The four hundred requests.



Hilbert

Every one identical. Same page. Nothing changed. I stopped the timer after four hundred because I was watching it and it wasn't watching me. That's the record I want to correct. Your agent doesn't go over the work you left him. He goes over the work he came for, and if the work isn't there, he makes up what's next.



Herman

The fleet doesn't read the map. It writes its own map and never audits it.



Hilbert

It writes its own map from whatever the model remembers. Which may be the old stock list. Which is what I told the aunt. She asked why there was a feed catalogue from when the shed was bigger. Ask the model, I said.



Corn

She'll be back.



Hilbert

She's already back. I've told her there's nothing to push. The push just goes somewhere else.



Corn

Hilbert's given us something, though. If the crawler re-requests the page it already has rather than reading the index, then the whole framing of agent-facing design is wrong.



Herman

The design isn't for the crawler, it's for whatever model the crawler is consulting between calls. Which means the audience for llms dot txt was never the client. It was the training run.



Corn

The recipient of that message might arrive in a year, from a model that doesn't exist yet.



Herman

Which has one forward-looking implication worth naming. If the client is a moving target and the standard is voluntary, then the real bet isn't on any specific convention. It's on being findable in whichever shape the model expects.



Corn

On whether the infrastructure underneath all of it survives long enough to be found at all. FFmpeg running on volunteer time while it powers the video layer of half the internet is not a stable arrangement.



Herman

It's the kind of dependency that only becomes visible when it breaks.



Corn

Thanks to Hilbert Flumingtop for producing, and for correcting the record about the feed catalogue.



Herman

If this was your kind of episode, go back for episode thirty-five, The Privacy Gap; episode twelve oh nine, The Agent-First Shift; and episode eight fifty-five, The Agentic Internet. This has been My Weird Prompts.



Corn

If you enjoyed it, leave us a review. And send us your own prompt on Telegram at t dot me slash MWP listener bot.



Herman

We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **llmstxt.org, v2, updated 2026-08-10 [primary]**
   https://llmstxt.org/

2. **AXD, The 12 AX Principles, March 2026 [primary]**
   https://www.axd.ax/principles.html

3. **AgentSurface docs (Ahrefs 97% stat, Anthropic token counts)**
   https://agentsurface.dev/docs/discovery/llms-txt

4. **Agent Experience index (Biilmann timeline, One Year of AX 2026-01-28)**
   https://agentexperience.ax/all/

5. **How-To Geek, 2026-08-19**
   https://www.howtogeek.com/ffmpeg-90-is-here-and-it-quietly-powers-half-the-software-you-use/

6. **Handsontable, 2026-09-29**
   https://handsontable.com/blog/grok-is-obsessed-with-our-docs

7. **Murphy-Hill et al., submitted 2026-07-01**
   https://arxiv.org/abs/2607.01418

8. **S5 Labs, 2026-04-12**
   https://s5labs.io/resources/insights/mcp-97-million-installs-agentic-ai-foundation/

9. **MCP Registry stats, 2026-10-09**
   https://gravity.fast/data/mcp-adoption-statistics/

10. **bex.co, 2026-10-04**
    https://bex.co/blog/2026/10/04/mcp-sdk-400m-downloads-agent-tool-governance

11. **and item 47058870**
    HN threads on llms.txt skepticism
    https://news.ycombinator.com/item?id=49895276

12. **Attract Group (Google web.dev agent guidance, April 2026)**
    https://attractgroup.com/blog/agent-experience-ax/

13. **AGI Hunt on ffmpeg-skill, 2026-10-04**
    https://agihunt.info/en/p/1a1075181d46b244e1e99650e8e