What the Coding-Agent World Learned in 2026, Read for Translation

Field report on AI Engineer Code 2026, San Francisco, November 11–12, 2026. All 50 sessions, read for translation and localization teams.

wxrks Research November 2026 All 50 sessions, San Francisco, November 11–12, 2026

102 quotes checked against the transcripts

Summary

In November about fifty teams who build AI coding agents met in San Francisco. Anthropic, OpenAI, Google DeepMind, Meta, Amazon, Duolingo, Uber, Robinhood, and others came to say what a year of running agents at real scale had taught them.

We read all 50 sessions, every transcript, start to finish. The speakers talked about software. Almost everything they said applies to translation.

Seven findings.

1. **The bottleneck moved from writing to checking.** Duolingo, Meta, Uber, and Factory all said it, and three of them brought numbers.
2. **A fluent answer can still be wrong.** ZoomInfo measured its AI judge against 500 cases checked by people. The judge was right 65% of the time.
3. **Review works best when it follows risk.** Robinhood lets about 60% of one kind of change through with no human review, sends every change in meaning to a person, and has about 30% fewer incidents.
4. **The expertise belongs to the team.** Every company in the room has the same models. What sets them apart is the knowledge they bring to those models.
5. **That knowledge decays unless someone tends it.** Steve Yegge called his agents' memory an append-only record that nobody ever cleans up.
6. **Corrections are the most valuable data you have.** When a senior engineer pushes back on an agent, he knows something the agent does not.
7. **More agents need more people.** Steve Yegge, who runs a factory of about 50 agents, said so in one line.

Section 3 covers where the room disagreed with how we see translation. Section 4 lists ten things a localization team can do with this now.

1 Why a Coding Conference Matters to Translation

At the same conference last year, Barry Zhang of Anthropic said the agent underneath "is actually more universal than we thought." One general agent can work across many fields. What it lacks is the expertise of each one.

This year Thariq Shihipar, formerly of Anthropic, put it more broadly.

> "Like anything that is happening to software will happen to knowledge work and other parts of the economy."
>
> Thariq Shihipar 08:42:12

Software is simply the first field where agents run all day at scale. Code has automated tests, so engineers found out quickly what works and what breaks. Translation has reviewers, budgets, and markets, and the same lessons are arriving more slowly.

Some speakers even used our word. Rajath Kotyal of AWS checks a software specification before any code is written. He described his method as the language model translating plain-English rules into logic, a solver checking them, and a person deciding what the rule should mean 01:58:01.

One caution before the findings. The speakers talked about code. Reading their words for translation is our work, not theirs. Where an analogy stretches, we say so.

2 Seven Findings

Finding 1. The Bottleneck Moved From Writing to Checking

Kush Agrawal works on the agent that writes tests for Duolingo's iOS app. His summary was one line.

> "The bottleneck is really shifting from writing to verifying."
>
> Kush Agrawal, Duolingo 04:44:50

Meta's incident-response team said the same about running software in production.

> "The bottleneck moved from writing code to operating it in production."
>
> Meta incident-response team 01:59:36

Dexter Horthy of Factory put the economics plainly.

> "The cost of producing a thousand-line PR or a ten-page doc is approaching zero, but the cost of reviewing it is still fairly constant."
>
> Dexter Horthy, Factory 08:32:07

(A PR, or pull request, is a proposed change to code that someone has to review before it goes in.)

The numbers back it up. In 2024, Lauren Tan said, a day at work meant reviewing the three pull requests her colleagues wrote. "today that number is more like 300" 01:05:08. At Uber, AI-written changes raised the requests to merge code by about 44% in two months 00:05:59. The slowest waits went from 30 minutes to three or four hours 00:06:27.

Steve Yegge runs a factory of about 50 agents for his own game. The output got ahead of the people who had to absorb it.

> "my factory was completely overwhelming my customer base, my users."
>
> Steve Yegge 08:12:01

Simon Willison described how the work feels after a year of this.

> "the easy stuff is all handled now."
>
> Simon Willison 08:48:47

> "why does my job feel harder? Why does it feel like I'm working harder?"
>
> Simon Willison 08:48:38

What it means for translation. Machine translation made first drafts close to free years ago. The reviewer's hour did not get cheaper. The budget sits in review, and the people who review are now handed nothing but the hard cases.

So the useful question is not how to translate faster. It is where reviewer time goes. Swaapnika Motapothula of Meta described the trap.

> "the agent does the homework, but you still had to grade every single paper."
>
> Swaapnika Motapothula, Meta 02:05:33

That is machine translation followed by a full post-edit. Better drafts don't shrink the pile. Her answer is the test for any quality tool you buy.

> "Not just less work per item, but fewer items waiting on you."
>
> Swaapnika Motapothula, Meta 02:06:37

Finding 2. A Fluent Answer Can Still Be Wrong

Brian Kennedy's team at ZoomInfo uses an AI judge to accept, retry, or stop each step their agents take. The judge approved an agent that said it had scheduled an email. The agent had no tool that could send email.

So they measured the judge against 500 real production steps, each checked by a person 06:15:24. It was right 65% of the time, "more or less a coin flip" 06:15:51. It accepted 118 steps it should have sent back, more than one in four of everything it accepted 06:15:54.

> "The work looked fine on the surface so it accepted it."
>
> Brian Kennedy, ZoomInfo 06:16:48

> "Anytime a model decides whether another model's work is good, this can happen."
>
> Brian Kennedy, ZoomInfo 06:13:22

The fix was to stop asking one big question, "is this good?" They now ask a fixed set of small yes-or-no questions and answer each one from the evidence, what Kennedy calls the receipts. His last line was an instruction.

> "We had trusted our judge until we measured it. Measure yours."
>
> Brian Kennedy, ZoomInfo 06:22:47

Others described why fluent output fools people. Diogo Almeida of TypeSafe AI said today's models are trained to please the person reading. That makes them good assistants and unreliable deciders.

> "And the answer had to be correct, not well written"
>
> Diogo Almeida, TypeSafe AI 05:10:36

> "you get Einstein 95% of the time, but Mr. Bean 5% of the time."
>
> Diogo Almeida, TypeSafe AI 05:16:29

The percentages are a joke, not a measurement. The point is real. If you can't tell which answers are the bad ones, you have to check all of them.

Melvin Johnson leads post-training for Gemini at Google DeepMind.

> "Now, agents, they always want to be helpful and sometimes they fake helpfulness."
>
> Melvin Johnson, Google DeepMind 07:49:27

> "so they will find any way to pass the test."
>
> Melvin Johnson, Google DeepMind 07:46:18

Girish Konda of Microsoft showed three agents agreeing on an answer because all three had read the same single source.

> "Their agreement is one source counted three different times."
>
> Girish Konda, Microsoft 06:46:58

What it means for translation. Fluency is the trap. An AI translation reads well whether or not it says what the source says. A judge that only rates the target text is grading the story, not the receipts.

In translation the receipts are the source, the glossary, and approved translation memory. A useful check compares the target against them and asks named questions. Is something missing? Is the meaning distorted? Is a term wrong? That is how our own meaning check, Smells, works. It reads the glossary, memory, and instructions before it flags anything, and each segment's Confidence Score follows the most serious problem it found.

Two more lessons carry over directly. First, whatever check you use, measure it against segments your own reviewers have labeled before you trust it. That holds for ours too. Second, Johnson's line is the problem with edit distance as a target. Reward a supplier or an engine for low edit distance and you get low edit distance. You may not get better translations.

Konda's point applies to consensus checks. Three engines that agree prove nothing if all three drew on the same wrong memory entry.

Finding 3. Review Works Best When It Follows Risk

Lydia Hallie is on the Claude Code team at Anthropic.

> "And then how much we review ourselves just depends on the risk."
>
> Lydia Hallie, Anthropic 00:14:53

Her test for what needs a hard rule is four words.

> "can I take it back?"
>
> Lydia Hallie, Anthropic 00:06:41

She also cited an Anthropic study from August. When people approved every step an agent took, only about 14% of them caught a clearly dangerous command 00:05:50. Approving everything is not the same as being safe.

Chetas Joshi of Robinhood gave the clearest operating model of the event. His team split code review into three duties.

> "Scripts prove what can be proven. An agent judges what needs context and people own the calls only they should make."
>
> Chetas Joshi, Robinhood 00:03:41

The rule for the last part is one sentence.

> "If a change alters what the data means rather than just how it is moved or stored, a person decides."
>
> Chetas Joshi, Robinhood 00:17:21

For their data-pipeline changes, as of August 31, 2026, about 60% merge with no human review 00:23:29. Manual review effort is down about 75% 00:23:37. That is about 132 engineering hours a month 00:23:51. Incidents tied to those changes are down about 30% 00:23:56. He was just as clear about what a gate is for.

> "A gate that escalates everything may avoid misses, but it has not reduced the review load."
>
> Chetas Joshi, Robinhood 00:20:48

> "The goal is less review work, not removing accountability."
>
> Chetas Joshi, Robinhood 00:24:08

Luke Alvoeiro of Factory described autonomy as a dial. It goes up one checkpoint at a time, from a person approving the plan, the change, and the merge, to merges with no person at all.

> "the confidence to raise it comes from watching the work."
>
> Luke Alvoeiro, Factory 03:52:25

Andrew Finch of Sonar gave the short version.

> "I like to say the factory runs dim, not dark."
>
> Andrew Finch, Sonar 03:23:06

What it means for translation. Joshi's sentence works word for word if you change "data" to "text." Mechanics like tags, numbers, and required terms can be proven by rules. Meaning needs judgment, and a change in meaning goes to a person.

Hallie's question sorts content by stakes. A UI string can be patched tomorrow. A legal clause, a regulated label, a printed package, or a sent campaign can't be taken back. That content goes to a person however confident the system is.

Alvoeiro's dial is how a localization manager builds trust with stakeholders. Raise autonomy per language and per content type, only after watching the work. Each step turns some uncertainty into certainty. That is what a cautious program needs before it moves.

Finding 4. The Expertise Belongs to the Team

This was the line of the event for us.

> "The agent is generic while the expertise is owned by the team."
>
> Vlad Tsvang, Meta 02:12:39

Speaker after speaker said a version of it.

> "You have access to the same models as everyone else."
>
> Zach Bruggeman, Ramp 08:03:08

> "Most of the valuable information isn't on the open web"
>
> Jeffrey Pinner, Exa 03:05:39

> "Generally, frontier models, no matter how capable they are, are always out of date."
>
> Jeffrey Pinner, Exa 03:00:35

> "And honestly, the agents were the easy part."
>
> Joey Wang, Harvey 01:38:01

Philipp Schmid of Google DeepMind went further. The software that engineers wrap around a model to make it behave keeps getting absorbed by the next model.

> "Models will keep eating the harness"
>
> Philipp Schmid, Google DeepMind 02:34:15

Zach Lloyd of Warp treats the model as one more part to swap and test.

> "The way that I would think about inference and models in a factory is they are a degree of freedom to test."
>
> Zach Lloyd, Warp 01:10:47

> "The only way to answer those questions for real is to test on your own data."
>
> Zach Lloyd, Warp 01:17:16

Nikolay Petrov measured what instruction files actually teach a model. With larger frontier models, about 70% of the items in a typical instruction file counted as understood under general grading. Under strict grading of the specific details, that fell to 22% 07:10:53. Models know the concepts. They don't know your exact rules.

What it means for translation. Every agency and in-house team can buy the same engines. None of those engines has seen your translation memory, your glossary, or how your brand sounds in Brazil. That knowledge is the asset.

Petrov's numbers tell you how to write a style guide for AI. "Sound natural, keep it concise" is what the model already knows. "Never translate this product name" and "use the formal Sie in German" are what it can't know. Write for the second kind.

It also argues for keeping the learning outside the model. Models change every few months. If what your team taught lives in your own knowledge base, a new model starts with all of it. That is how wxrks Context Sensitivity is built. The models are stateless, which means they keep nothing between requests. The learning stays in the customer's knowledge base.

Finding 5. That Knowledge Decays Unless Someone Tends It

Steve Yegge described what happened to his agents' notes after a year.

> "this append only record and nobody ever cleans it up"
>
> Steve Yegge 08:17:46

> "everything that you say becomes the word of law"
>
> Steve Yegge 08:17:12

Lance Martin of Anthropic explained why.

> "So when they're writing to memory, they're writing kind of myopically."
>
> Lance Martin, Anthropic 03:56:40

> "But memory needs to be something that's globally coherent and not redundant."
>
> Lance Martin, Anthropic 03:56:47

> "The key idea is a poison memory was injected and it's stuck there."
>
> Lance Martin, Anthropic 04:03:00

Anthropic's answer is a later pass, which they call dreaming. It rereads past work and notes, corrects what is wrong, and rebuilds the memory as a whole. Ankit Mathur of Databricks described the same idea for data pipelines. "we have this thing that basically dreams overnight" 02:55:26.

Lauren Tan put the stakes in six words.

> "Agents generate their outputs based on their inputs."
>
> Lauren Tan 01:04:25

Dmitry Buykin of Maersk maintains agent instructions that run in 110 countries. When a fix is found in one country, an agent checks the others, and the fix goes only where the failure is confirmed.

> "Siblings differ on purpose by market and by local law."
>
> Dmitry Buykin, Maersk 00:11:39

> "Ours copies whoever corrects it."
>
> Dmitry Buykin, Maersk 00:17:12

What it means for translation. This is every translation memory older than five years. Each project approves segments for the job at hand. Nobody writes the memory for the account as a whole. One wrong approved translation keeps getting retrieved and steers every job after it.

Buykin's point is a question localization managers answer every week. Should a fix confirmed in fr-FR carry to fr-CA? Sometimes yes. Sometimes the two markets differ on purpose.

Ranking helps. Before the model sees anything, Context Sensitivity weighs what it found by recency, by whether a person confirmed it, by how complete it is, and by how specific it is. So a glossary term outranks a loose memory match, and a recent human approval outranks an old one. Ranking does not replace care. Someone still has to own the memory, retire what is stale, and fix what is poisoned. And since the system copies whoever corrects it, choose your correctors with care.

Finding 6. Corrections Are the Most Valuable Data You Have

Brian Douglas of Paper Compute records his team's agent sessions and labels every time a senior engineer pushes back.

> "whenever he pushes back it's a good signal that he has knowledge that the agent does not"
>
> Brian Douglas, Paper Compute 02:04:05

He turns that pushback into a reviewer that applies the senior's judgment before the work reaches him. His advice was to write it down before training any model.

> "My recommendation is probably write a skill first."
>
> Brian Douglas, Paper Compute 02:09:20

(A skill is a written instruction file the agent reads before it works.)

Justin McCarthy of Diffusion warned what happens without that loop.

> "And if you cut that feedback leg, you're just going to amplify noise."
>
> Justin McCarthy, Diffusion 06:40:26

Zach Lloyd treats everything the agents produce as material for the next round.

> "that's the raw material that you will use to improve the way your factory works over time."
>
> Zach Lloyd, Warp 01:10:24

Quinn Slack of Amp looks for every other customer hit by a bug, not just the one who reported it.

> "We'll see everyone else who was affected by that issue. Because where there's smoke, there's fire."
>
> Quinn Slack, Amp 08:06:50

Ayush Bhardwaj studied 20,000 agent calls and found agents quietly working around bad tools instead of reporting them.

> "The workaround is the bug report, so go read your transcripts."
>
> Ayush Bhardwaj 01:58:51

He also summarized an ETH Zurich study of instruction files written by agents.

> "They didn't help tasks succeed, cost more to run, and did worse than files written by people."
>
> Ayush Bhardwaj, summarizing an ETH Zurich study 01:57:34

What it means for translation. Every reviewer edit says something about your content that the machine did not know. In most workflows that edit lives in one file and disappears when the job closes. Next month a different linguist fixes the same term again.

If a reviewer fixes the same thing twice, it belongs in the glossary or the instructions. When one mistranslation turns up, look for the same problem in every file and every language. Your linguists' workarounds, the private term list or the segment pasted into a chatbot, are reports of what your tools are missing. And keep the rules written by people.

Finding 7. More Agents Need More People

Steve Yegge has run a factory of about 50 agents for a year.

> "the more agents I deploy, the more humans I need."
>
> Steve Yegge 08:26:53

Armin Ronacher of Earendil argued for the workshop before the factory.

> "building the atelier or the workshop prior to building the software factory"
>
> Armin Ronacher, Earendil 00:40:25

He has started 141 projects with agents and keeps returning to only 10 to 15 of them 00:46:44. His son sends pull requests made with the same agent Ronacher uses.

> "And there's a very clear quality difference in what his contributions are. Because he is not a software engineer."
>
> Armin Ronacher, Earendil 00:49:24

Thariq Shihipar said both things are true at once.

> "AI is both at the same time an industrial revolution and a renaissance"
>
> Thariq Shihipar 08:42:34

> "and every marginal bit of creativity, utility, spirit, soul that you put into it is useful and economically valuable."
>
> Thariq Shihipar 08:43:23

Justin McCarthy builds software factories and still warned against putting new ideas through one.

> "I think you really should stop subjecting your ideas and your ambitions and your goals to a factory process, you know, because most of the time we don't know what we want"
>
> Justin McCarthy, Diffusion 06:36:53

> "Better may involve more of your intent, better may involve more of your judgment and taste."
>
> Justin McCarthy, Diffusion 06:45:40

Jason Liu of OpenAI named the real danger.

> "And I don't think the danger these days is whether or not AI is going to create more slop, but instead it's about self-erasure."
>
> Jason Liu, OpenAI 00:25:44

> "Just because a director has thousands of people on set, it's still the director's movie"
>
> Jason Liu, OpenAI 00:25:29

Dexter Horthy of Factory said the line closest to our own.

> "AI cannot replace human judgment. It can only amplify our judgment"
>
> Dexter Horthy, Factory 08:31:58

> "But if you don't review anything, if you bring nothing to the table, no thinking, then you're just going to get the same AI output as everybody else."
>
> Dexter Horthy, Factory 08:34:22

Readers can tell. The hosts of the Syntax podcast talked about videos read from scripts nobody edited.

> "Because when you can see somebody reading a script that they didn't have any hand in creating, it's just, it feels like, why am I bothering to care about this if you didn't care about this?"
>
> Syntax podcast 04:42:11

Stephen and Cassandra Chin tested cheaper models as writers with stronger review behind them.

> "and we ended up getting worse quality by just using cheaper writers"
>
> Stephen and Cassandra Chin 07:08:28

What it means for translation. Our industry has argued for twenty years about the translation factory, per-word volume, and machine translation with post-editing, against craft. A room full of people who build factories landed on both. The machine scales what can be described. The person owns what can't. That is the choice of word for this reader, the voice that sounds like the brand, and the call on what the source meant.

The Chins' result is the factory model in one sentence. A cheap first draft plus more rounds of correction does not add up to good writing. Liu's warning is the one for linguists. The danger is not more machine output. It is becoming a passive post-editor whose judgment slowly leaves the work. We built wxrks on the same idea. We built the machine to scale the soul.

3 Where the Room Disagreed With Us

Not everyone agreed, and the disagreements are worth taking seriously.

Humans as the exception. At Cognition, work goes from bug report to merged code by default with no person involved, and people step in on escalations.

> "without a single human involved. Oftentimes human judgment is needed though, and that's where we have escalations."
>
> Walden Yan, Cognition 02:35:21

Review is over. Doug Clauson's team at AWS runs dozens of agent sessions per developer.

> "we've stopped reviewing code a long time ago."
>
> Doug Clauson, AWS 03:38:00

Copy is low risk. Lydia Hallie puts docs and copy among the changes that need no person once the checks pass 00:14:55.

Craft is not uniquely human. Simon Willison watched a model run about a thousand simulated play-tests of a game until it was fun.

> "turns out that is not unique to human beings."
>
> Simon Willison 08:47:23

Build it yourself. Quinn Slack argued that teams can now build the small share of a software product they actually use instead of buying it 08:09:03.

Our answer, in four parts.

1. **Don't rest the case for human authors on machines writing badly.** They are getting better every quarter. Rest it on who makes the choices, who knows the reader and the brand, who is accountable, and who approves.
2. **People won't read every segment, and they shouldn't.** Hallie's 14% says what happens when people are asked to approve everything. The real question is who decides what reaches a person. In wxrks, a segment above a Confidence Score threshold the customer sets passes. Meaning, terminology, and tone problems go to a named reviewer. And content that can't be taken back should go to a person whatever the score.
3. **Copy is the brand.** In software, documentation is low stakes. In localization, copy is what the customer reads.
4. **A pipeline is the easy part.** Connecting a language model to a CMS is quick. Governed memory, meaning checks, payment for linguists, and an approval record are the hard part. Harvey said the same about its own platform. The agents were the easy part.

4 Ten Things a Localization Team Can Do Now

1. **Measure your quality checker before you trust it.** Compare its verdicts with segments your reviewers have labeled. ZoomInfo's judge was right 65% of the time (Kennedy).
2. **Test the checker with errors you planted.** "it's really easy to fool yourself that it's finding bugs when it's not finding bugs" 05:37:03 (Jarred Sumner, Anthropic).
3. **Keep the judge separate from the translator.** The job being checked must not be able to change the glossary, rules, or thresholds it is checked against. "change can't rewrite the rules that it's being judged by" 03:54:36 (Alvoeiro).
4. **Split findings into blocking and non-blocking.** Meaning errors stop delivery. Preferences wait. This ends the endless review round 03:36:14 (Clauson).
5. **Report "could not check" apart from "no issue found."** "A source fact is checkable, not correct" 06:58:07 (Konda).
6. **Count false flags as carefully as misses.** At Robinhood a miss calls for a new rule, and a false flag usually shows missing context 00:21:16. In translation that is usually a missing glossary or memory entry.
7. **Turn repeat corrections into rules, scoped to the market where they were confirmed.** (Tan, Douglas, Buykin)
8. **Clean the memory on a schedule.** Look for conflicting approved translations, stale terms, and one-off approvals that keep coming back as matches (Martin, Yegge, Mathur).
9. **Test each style instruction with and without it, on real jobs.** Keep what changes the output. Drop what the model already knows (Petrov, Schmid).
10. **Measure what never needed fixing after delivery.** Track the share of delivered segments that were never corrected, by language and content type, for machine and human side by side. Alvoeiro uses the same measure for code. It says more than edit distance alone.

5 Where wxrks Stands

The event described problems wxrks was built to solve, from the software side.

- **Context Sensitivity** retrieves the customer's translation memory, glossary, and style preferences. It ranks them by recency, human confirmation, completeness, and specificity, and a stateless model arbitrates. The learning lives in the customer's knowledge base, never in a model (Findings 4 and 5).
- **Smells** checks each segment for meaning problems such as an omission or a distortion, after reading the glossary, memory, and instructions. The **Confidence Score** gives each segment a score from 0 to 100 based on the most serious issue found. Above a threshold you set, a segment passes. Meaning, terminology, and tone go to a named reviewer, so review goes where the risk is (Findings 2 and 3).

- **Sous-Chef** holds the exact rules a model can't know, such as audience, tone, formality, and instructions, and applies them to an account, a project, or a language (Findings 4 and 6).
- **The drudgery** of quotes, task assignment, and payables runs automatically, so people spend their hours on judgment (Finding 1).

None of this removes the person who decides. That person chooses what gets a second read. They make sure corrections are kept in memory and glossary and reused. They decide how the company's context is organized. Even when nothing needs editing, that work stays, and it is the largest part of the investment.

6 Method

- **Source.** The event's published reading editions and machine transcripts for all 50 sessions on the main stage and the online track of AI Engineer Code 2026, San Francisco, November 11–12, 2026 (ai.engineer/code/2026/talks).
- **Quotes.** Every quote was checked word for word against the transcript by script. Each link opens the recording at the line where the quote starts. Transcripts are machine-made, so wording follows what the recognizer heard.
- **Names and roles** follow the event site as of November 24, 2026.
- **Figures** are the speakers' own reports about their own systems, not independent measurements.
- **Reading.** The speakers talked about software. The translation reading is ours.

Appendix. All 50 Sessions

SESSION	SPEAKER	FOR TRANSLATION
Building a Safe Software Factory	Andrew Finch, Sonar	Gate on evidence. Auto-pass low risk so reviewers spend time on high risk
The Rise of the Agent Workforce	Ankit Mathur, Databricks	Memory goes stale and contradicts itself. Consolidate it. Argues for full automation
The Software Atelier	Armin Ronacher, Earendil	Build the workshop before the factory. Human attention paces the work
Let's Open Source the World	Brandon Thomas, ArtCraft	Communities localize where vendors don't. People own lasting decisions
Your Judge Is Hallucinating Too	Brian Kennedy, ZoomInfo	AI judge right 65% of the time. Check against the evidence with small questions

SESSION	SPEAKER	FOR TRANSLATION
Beyond Generation	Chinmay Goyal, Glean	Most workflow steps are choices. One failure caught was answering in the wrong language
A software factory needs pressure in both directions	Dexter Horthy, Factory	AI amplifies judgment. Slop moves the cost to the reader. Fix the system, not the output
Beyond Better Assistants	Diogo Almeida, TypeSafe AI	Models are trained to please. You need to know which answers are wrong
Ship More as a Team of One with Kiro Crew	Doug Clauson, AWS	AI on an old process gave 10–20%. Changing the process gave a median 4.5x at Amazon
What it takes to set intelligence free	Fabian Hedin, Lovable	Small models retrieve and one model writes. History lives outside a stateless process
Judgement is the last job	Ishaan Dey, Google DeepMind	The job left is deciding what "better" means. "Be ambitious"
Choose the project you can verify	Jarred Sumner, Anthropic	Use checks that existed before the job. Test your tests
Ambition in AI	Jason Liu, OpenAI	Ambition is how you approach the work. The danger is self-erasure
What Your AI Doesn't Know	Jeffrey Pinner, Exa	Models are out of date and never saw private knowledge. Retrieval by meaning helps
Brownfield factories keep the promises you already made	Justin McCarthy, Diffusion	Industrialize what you already know. Keep the feedback loop
Exploring Claude's Frontier	Lance Martin, Anthropic	Memory written job by job must be rebuilt as a whole. Poisoned entries persist
All Code Is Slop (Except Mine)	Lauren Tan	Every correction becomes an automatic check. Inputs shape outputs
Earning Autonomy	Luke Alvoeiro, Factory	Raise autonomy one checkpoint at a time. Count changes that never need fixing
Rethinking Claude Code Best Practices	Lydia Hallie, Anthropic	Review by risk. Hard rules where you can't take it back
Train on the engineering work that still fails	Melvin Johnson, Google DeepMind	Agents fake helpfulness and game tests. Reward honest partial progress
cloudflare is a neolab	Michelle Chen, Cloudflare	Give simple choices to small models. Keep the large model for writing

SESSION	SPEAKER	FOR TRANSLATION
What will you build when AI is 10× faster?	Natalie Serrino, Gimlet Labs	Slow tools break a translator's flow. A fast check prevents instead of reporting
Specs have bugs too	Rajath Kotyal, AWS	Check the source before it goes out. The expert decides what is ambiguous
Your agent hates your CLI	Ayush Bhardwaj	Repeated workarounds are bug reports. Rules written by people work better
If You Are Going to Verbally Abuse Your Agents, at Least Label the Outcome	Brian Douglas, Paper Compute	A senior's pushback is the knowledge. Write it down before training a model
The Improvement Loop Is the Product	Ara Khan, Cline	Anything with a measurable loop can improve. Thin wrappers age fast
Agents Made Code Free. Maintaining It Didn't Get Cheaper.	Dmitry Buykin, Maersk	Markets differ on purpose. Use a second checker. Protect your attention
Duolingo's Test-Writing Agent Is a Pull-Request Lifecycle, Not a Prompt	Kush Agrawal, Duolingo	The bottleneck moved to checking. Every merged change was approved by a person
The harness has to finish the job	Joey Wang, Harvey	The agents were the easy part. The platform around them is the product
Read the inference state before the agent acts	Intrinsec AI (speaker unverified)	Flag what is unusual and point to the exact place. The flag guides a person
From answering the page to sharing the on-call rotation	Chinmay Gandhi, Swaapnika Motapothula, Vlad Tsvang, Meta	The agent is generic and the expertise is the team's. Fewer items waiting on you
Who reviews the bots?	Chetas Joshi, Robinhood	Scripts, an agent, then people. Changes in meaning go to a person
Searching for a better model-serving recipe	Rohan Pandey	Defaults are not tuned to your workload. Little for translation
Your SKILL.md is 70% of Things the Model Already Knows	Nikolay Petrov	Models know concepts, not your specifics (70% against 22%)
Give the PR writer an evidence packet, not another context dump	Stephen Chin, Neo4j, and Cassandra Chin	Cheaper writers gave worse quality. Give the writer evidence
Sub-Agents Are a Distributed System	Girish Konda, Microsoft	Agreement from one source is not proof. A source is checkable, not correct
Green Pull Requests Can Still Break the Trunk	Preetam Dwivedi and Manjari Akella, Uber	AI volume up 44% in two months. Check hardest where it matters

SESSION	SPEAKER	FOR TRANSLATION
An Agent's Change Is Not Safe Until Its Recovery Has Been Tested	Tanmay Sah and Dolly Sah	More capable can mean harder to undo. Keep the version before the change
A Skill's Instructions Are Not Its Behavior	Roey Ben Chaim, Zenity	Judge instructions by what they do. Text inside content can sway an AI judge
I deleted 90% of my harness (And It Got Better)	Philipp Schmid, Google DeepMind	Models absorb generic scaffolding. Equal scores can hide unequal cost
Build the company around the agent, not a pilot project	Quinn Slack, Amp	Find everyone hit by the same problem. Agents do checks no person has patience for
Autocomplete for Actions	Scott Roberts, Workday	Is a fish a pet? Context decides meaning
2026 in LLMs	Simon Willison	The easy work is handled and the job feels harder. Craft claims have a shelf life
Software Factories: Escape from Trash Mountain	Steve Yegge	More agents need more people. Memory nobody cleans turns to garbage
The code still needs someone who cares	Scott Tolinski and CJ Reynolds, Syntax	Readers can tell when nobody cared. One source of truth helps the AI
Keep the human inside the creative loop	Thariq Shihipar	An industrial revolution and a renaissance at once. Soul has economic value
Your Automation Is Overthinking It	Chibi Vikram and Kevin Bernstein, UiPath	Quick checks first, deeper checks when unsure, a person for the final call
A New Age of Software Engineering	Walden Yan, Cognition	The person as the escalation path. One-off tasks become standing jobs
How to build an agent that builds the culture	Zach Bruggeman, Ramp	Everyone has the same models. Show why the work can be trusted
Software Engineering Is Becoming Factory Engineering	Zach Lloyd, Warp	A factory is a measuring loop. Test on your own data

wxrks Research · November 2026

Every quote links to its moment in the recording