AI-ASSISTED DEVELOPMENT FOR WONDER BY DELOITTE

Day 1.
Kickoff & live demo.

A five-week program to make agentic development the default across Wonder teams.

LAZER • PRESENTS

wonder
BY DELOITTE

ABOUT LAZER

We went through this transition ourselves, first.

LAZER TECHNOLOGIES

200+

person product studio

Internal AI rollout conducted before most of the industry started.

01 **We built it for ourselves, at scale.**

We ran a structured AI rollout across every engineering domain inside Lazer, including picking internal champions, codifying standards into workflows, and iterating on tooling and processes as we learned what actually worked.

02 **We've helped other teams make the same shift.**

Since then, we've walked other teams through it, from early-stage startups to enterprise clients. The scale and the stack are different every time, but the underlying transformation is remarkably similar.

03 **This program is a direct product of that work.**

What you're going to see in the next five weeks is not generic training material. It's the playbook we wrote for ourselves and have since refined on real client engagements.

YOUR TEAM FOR THE ENGAGEMENT

Who you'll actually be working with.

D

Dave

LEAD TRAINER

In every session

Dave is a senior engineer at Lazer and has been shipping software for over 15 years. Before joining us, he spent three and a half years as Lead Instructor at Lighthouse Labs, so he is as comfortable teaching as he is building. He runs every session in this program and is your main point of contact throughout.

L

Luca

DELIVERY MANAGER

Available for logistics, admin, and program questions

Luca is already working with the Wonder team on our other engagements, so you have met him before. He's supporting this program on scheduling, logistics, and anything else on the delivery side, so you can focus on the sessions and not on coordination.

ALSO AROUND FROM LAZER

Megha, Head of Fintech, **Laura**, Delivery Manager support, and **Andrew**, Head of Engineering, may drop in on a session here and there.

TODAY'S SESSION

Here's how today is going to run.

Four parts, and questions are welcome throughout.

01

Why now

A quick look at what's actually possible with agentic development today.

02

Live demo

Dave working in Cursor, live.

03

Cursor 101

The modes, the models, and the mistakes we see people make early on.

04

The program

What the next five weeks will look like, plus a bit of homework.

PART 1 · WHY NOW

Agentic development is already a production reality.

01 A new class of engineer is already here.

They explore an unfamiliar codebase in minutes instead of days. They generate validated test suites in a fraction of the time it used to take. They ship features at a pace that, a year ago, would have required a team twice the size. Most orgs may already have one or two of them. The goal of this program is to make that capability common across all Wonder teams.

02 This is not early-adopter territory anymore.

Most of the Fortune 500 is already shipping with AI tooling in the loop. The conversation in engineering orgs has moved on from whether to adopt this and is now about how *fast* and how *deep*. The gap between teams that have figured it out and teams that haven't is widening every quarter, and catching up from behind is harder than keeping up from the front.

PART 2 · LIVE DEMO

Enough talk.
Let's see it.

CURSOR · LIVE

PART 3 · CURSOR 101

So what is Cursor?

WHAT YOU JUST SAW

The agent read the codebase first.

Before writing a single line of code, it grounded itself in what already existed. It wasn't guessing from a blank slate.

Every change created a rollback point.

That happened automatically. You can undo a single file or the whole session without any manual bookkeeping on your side.

Every diff was reviewed before it landed.

Dave rejected one on purpose to show you how it works. Nothing gets into your branch unless you actively accept it.

WHAT CURSOR IS

A full AI-native IDE.

Cursor was built from the ground up around agentic workflows, rather than an AI assistant bolted onto an existing editor.

Codebase-aware by default.

It reads your entire project, not just the file you have open, and does its best to follow the conventions it sees already in place.

Four modes for four levels of autonomy.

Tab, Ask, Agent, and Plan. You pick the level of help you want depending on the task. We'll go through all four on the next slide.

You stay in control throughout.

You accept, edit, or reject every diff. Nothing lands in your branch unless you actually say yes to it.

CURSOR 101 · PART 1

The four modes.

Quick dive, not a deep one. Week 1 covers each in detail. For today: enough to start playing.

TAB

FOR

Autocomplete

WHEN

You're already typing and you just want help finishing the thought.

ASK

FOR

Read-only exploration

WHEN

There's something in the code you don't understand yet, and you want to figure it out before changing anything.

AGENT

FOR

Build and execute

WHEN

You know roughly what you want to happen and you're comfortable letting it go do the work.

PLAN

FOR

Research and plan first

WHEN

The task is big or ambiguous enough that you want a plan locked in before any code gets written.

CURSOR 101 · PART 2

Picking a model.

Start on Auto. Override with intention once you have a feel for it.

AUTO

A reasonable default that gets you moving without having to think about model selection on day one.

One thing worth knowing: Auto optimizes for cost, which isn't always your optimization target. On harder or higher-stakes tasks you'll often get better results by picking a model deliberately rather than letting Auto route for you.

CLAUDE

SONNET 4.5 · OPUS 4.6

Your main coding work.

The strongest models out there right now for reasoning and following instructions. Use Opus when the problem is genuinely hard, and Sonnet for pretty much everything else.

GPT-5.4

OPENAI

A second perspective.

Trained differently than Claude, so it tends to catch things Claude misses and vice versa. A good one to reach for when you're stuck on tricky debugging.

COMPOSER 2

CURSOR

Speed over depth.

Cursor's in-house model, tuned for fast agent loops. Best for simple edits and quick iteration rather than deep reasoning on a hard problem.

CURSOR 101 · PART 3

Mistakes we see people make early on.

01 Don't skip the code review.

AI gives you a fast first draft, but your name is on the PR. Read every line of the diff the same way you would read a teammate's. The bar doesn't drop just because it came out of a model.

02 Don't hand it the keys without thinking.

Respect tool permissions. Read what the agent wants to run before you approve it. Don't connect random MCP servers; treat them the same way you'd treat installing any new dependency.

03 Don't over-optimize model choice.

Defaulting to the cheapest model for everything is a trap, and defaulting to the most expensive is also a trap. Start on Auto to get moving, and override when you have a real reason to.

04 Don't dump everything into context.

More context is not always better. Irrelevant files add noise, confuse the model, and cost more for worse results. Be deliberate about what you pull into the conversation.

05 Don't stare at the spinner.

Agents run in the background, which means you get your time back while they work. Use the wait to plan the next task, review the last diff, or kick off a second agent on something parallel.

IT ISN'T JUST FOR WRITING CODE · 1 OF 2

Other things engineers can do with it.

Most of the value isn't in "generate me a function." It's in the things you already do every week that are slow, tedious, or easy to get wrong.

EXPLORING AN UNFAMILIAR CODEBASE

"Walk me through how authentication flows from the login endpoint to the session middleware. Show me the files involved and flag anything unusual..."

DEBUGGING A FLAKY TEST

"This test fails intermittently. Read the test, the function under test, and the last 10 commits touching either file. What are the most likely causes..."

REFACTORING

"Rewrite this function to eliminate the nested callbacks. Keep the public interface identical. Add a brief comment explaining the change..."

IT ISN'T JUST FOR WRITING CODE · 2 OF 2

And it isn't just for engineers, either.

PMs, designers, and data folks on your team can get a lot out of the same tool. They just tend to ask it different questions.

PMs

WRITING A GROUNDED TICKET

"Look at the checkout flow in src/checkout. I want to add a gift message field. Draft a ticket with the files that would need to change, rough complexity, and open questions..."

INVESTIGATING A PROD ERROR

"Here is a stack trace from Sentry: [...]. Which file raised it, what does that code do, and what is the likely user-facing impact..."

DESIGNERS

FEASIBILITY CHECK

"Look at our Button component. Can it support an inline loading spinner without breaking existing usage? What would change..."

COMPONENT INVENTORY

"What variants does the Card component support today? Show me how it's used across the app..."

DATA SCIENTISTS

PIPELINE WALKTHROUGH

"Starting from the ingestion service, walk me through how raw events become features in the feature store. Call out any transformations that look suspicious..."

PART 4 · THE PROGRAM

The next five weeks.

WEEK 01

Foundations

The core building blocks. Modes, prompting, rules files, how context works, and how to pick a model for the job.

WEEK 02

Team patterns

Moving from individual productivity to shared practice. Codebase rules, MCP integrations, custom agents, and review workflows the whole team can follow.

WEEK 03 · 04

Embedding

We pair with your engineers on real sprint tickets over two weeks. We keep decision logs, and end each week with a short readout so patterns get shared across the team.

WEEK 05

Consolidation

Shaped by where the team lands after the first four weeks. We hold this week loose on purpose so we can zoom in on whatever still needs work.

A FEW THINGS TO KNOW ABOUT HOW WE RUN IT

Sessions are hands-on, so bring real work from your actual backlog rather than sandbox projects. Questions that feel obvious are usually the ones most people in the room are also wondering about, so ask them. Every session is recorded and posted back to the Teams channel the same day, and Dave is on Teams throughout the engagement if something comes up between sessions.

THE OUTCOME

What success looks like.

By the end of five weeks.

01 Agentic-first, by default.

Reaching for AI on real work becomes automatic rather than a conscious decision the team has to remember to make.

02 Every repo carries its own context.

Rules, conventions, and shared context live in the repo itself, so new hires inherit the team's accumulated knowledge just by opening the project.

03 Leverage across roles, not just engineering.

PMs, designers, and data folks on the team are getting meaningful value from the same tooling, instead of it being an engineering-only advantage.

04 The capabilities transfer.

Planning before generating, verifying before shipping, breaking big tasks into smaller ones, and knowing when AI genuinely helps versus when you should trust your own judgment.

WHY THIS MATTERS Tools will keep changing. The concepts you build in the next five weeks are what stays.

RESOURCES

Where to keep learning.

Quality over quantity. Check these weekly, not daily.

OFFICIAL

Start here.

Cursor docs and blog

cursor.com/docs · cursor.com/blog

Cursor changelog

Worth checking weekly. The product moves fast.

Model lab blogs

Anthropic and OpenAI engineering. Primary source for where the models are going.

COMMUNITY

High-signal only.

Hacker News

Cursor and AI-assisted dev threads are usually thoughtful

Andrej Karpathy on X

One of the best signals on where AI and software are going

Dave's picks

Dave will be sharing more throughout the program. Check the Teams channel.

EVENTS

In person.

Cursor Meetup Toronto

A local meetup with talks and a hackathon. If you're interested, details and signup are at the link below.

luma.com/11fprizv

BEFORE THE NEXT SESSION

Two things to do before we meet again.

01

Open Cursor on something real.

Pick a file on a real project that you didn't write yourself, and use Ask mode to have Cursor walk you through it. Pay attention to what it got right and what felt off. Bring both observations to the next session so we can talk through them as a group.

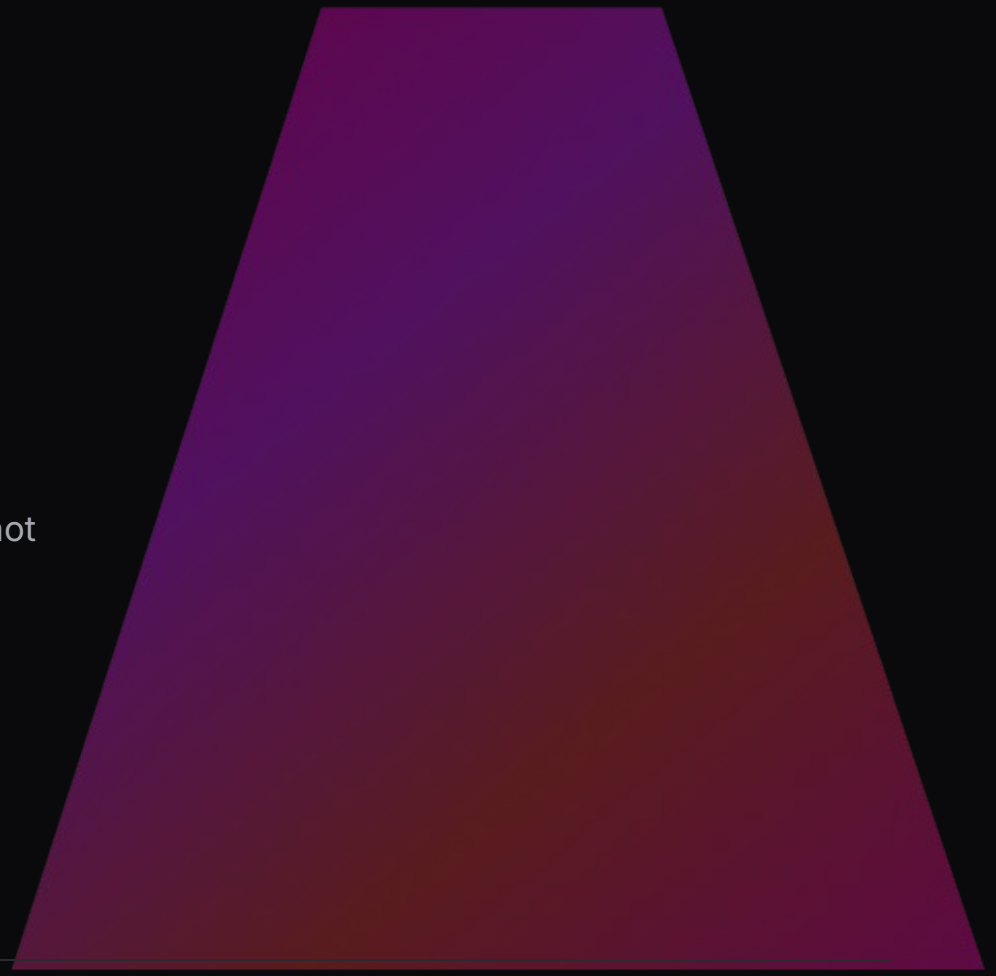
02

Complete the pre-learning survey.

It's part of the AI Readiness Assessment we're running alongside this program. Your answers directly shape what we focus on in Week 1, so it's worth doing honestly rather than what you think we want to hear. The link is in the Teams channel.

OVER TO YOU

Questions?



ASYNC CHANNEL IN TEAMS

For anything that comes up after today. Dave checks it throughout the engagement, not just during session time.

DAVE IS ON TEAMS

Quick questions, bouncing ideas, brainstorming an approach before you start, post-mortems on things that didn't work. Use him.