Product builders: no results yet.



+220%
growth in code changes to internal systems in a year

+36%
growth in changes that reached users

94%
of code changes opened by an agent

+40%
growth in major outages in a year

3×
merged code changes per engineer in 16 months

74→55%
staff satisfied with work in the internal survey

−39%
median time from idea to release cut by 39%

+70%
growth in time spent firefighting

Some developers, product managers and analysts become builders. The other roles become their support: data, review, hard cases, the platform.

Delivery really is speeding up: code gets written faster, and everyone notices. Quality and reliability lag behind: rollbacks, outages and tech debt grow. Discovery and analytics are not speeding up yet, and teams end up with empty backlogs. Analytics gets done partly or skipped.

Nobody has measured the effect on the business.

Inside: 5 pilots (Meta, Atlassian, Intercom, epilot, Uber), what breaks, what happens to analytics and how roles change.

Uber

Five pilots.

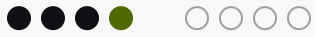
A builder is one person who comes up with a feature, designs it and builds it with AI agents. Only companies that described how they set up the pilot and what came out of it.

BEFORE



A team of 10–20: 7–14 engineers, a product manager, a designer, an analyst, a researcher, a data engineer

AFTER



A group of 3–5: 3–4 builders, usually former engineers, and one person who decides what to build. Designers and analysts in a shared team across groups

Meta's internal slide, published by Reuters

Meta

Partly rolled back

WHAT THEY DID

- Groups of 3–5 instead of teams of 10–20
- ~1,000 people from Reality Labs in the pilot

WHAT CAME OUT

- Code +220%, changes for users +36%
- Major outages +40%

Atlassian

Rolled back

WHAT THEY DID

- ~10 projects of selected builders
- Product managers, designers and engineers in one role

WHAT CAME OUT

- No figures published
- Fast first weeks, then a sharp slowdown

Intercom

Outcome unknown

WHAT THEY DID

- ~500 people in R&D, goal: twice the accepted code changes per person
- 94% of code changes created by an agent, designers ship their own fixes

WHAT CAME OUT

- Three times the changes per person in 16 months
- Bug queue −54%, idea to release −39%

epilot

Continuing with limits

WHAT THEY DID

- 60 days, 50+ people outside engineering given access to code
- Every change is tried on a copy of the product first

WHAT CAME OUT

- 40 of 629 changes came from 11 people outside engineering
- One pushed a change to production past the automatic checks

Uber

Outcome unknown

WHAT THEY DID

- A year of product managers prototyping with AI
- The prototype goes with the task description

WHAT CAME OUT

- “Two hours of prototyping replaced four weeks of discussion”
- Prototypes get taken for the finished solution

Left out: LinkedIn, Coinbase, Block and Ramp announced programmes without results; OpenAI's team simply grew from 3 to 7 engineers.

What breaks.

More code, same value

+220% / +36%

code / changes for users, Meta

- Meta: code +220%, changes for users +36%
- Harvard, 718 companies: code +30% after AI agents, finished tasks did not grow

Nobody decides what to build

1 : 30–40

product manager or designer to engineers, Atlassian

- Atlassian: everyone went off to write code, nobody set the direction
- It felt like one product manager or designer per 30–40 engineers instead of 10

Review cannot keep up

+40%

major outages, Meta

- Intercom: code review became the bottleneck, quality dipped
- Meta: major outages +40%, firefighting time +70%
- epilot: a change reached production without the automatic checks

People burn out

74% → 55%

satisfied with work, Meta

- Meta: staff satisfied with work fell from 74% to 55%
- Former managers asked to come back to management

Analytics for builders.



I see the fourth option more often than I would like.

Analytics falls behind.

My observations

01

Analytics gets done halfway

A builder ships a feature and does not check whether it helped. When they do run an A/B test, they often cannot set it up or read it properly.

WHAT OPEN SOURCES SHOW

B. Booking NOT A PILOT

80% of tests never checked in advance whether there were enough users to see an effect. Courses did not help; hints in the testing platform did

Uber Uber

making a prototype became easy; working out what it tells you did not

02

Builders don't want to do it

Builders do not want to dig into data themselves and go to an analyst. The load on analysts does not drop.

WHAT OPEN SOURCES SHOW

Intercom

the company president used to file a request with an analyst and wait. They built an agent that knows the company's data

Ramp

1,476 questions to the agent in 4 weeks, 66 in the data help chat

03

The agent has nothing to stand on

Without instructions and shared metric definitions the agent writes queries from scratch, numbers disagree, and the builder goes back to an analyst.

WHAT OPEN SOURCES SHOW

Anthropic NOT A PILOT

without instructions the agent was right at most 21% of the time, with them over 95%. Without upkeep, 65% a month later

Meta

moved its agent to ready, checked queries and bought a startup that builds a shared metrics layer

Sloppy code and look-alike products.

My observations

04

Code gets sloppy, debt piles up

Tests, docs and clean-up are skipped for speed. In 6–12 months every change takes longer and costs more.

WHAT OPEN SOURCES SHOW

Intercom

when AI started writing most of the code, code quality fell, then they pulled it back up

Atlassian

fast first weeks, then a sharp slowdown

Meta

major outages +40%, firefighting time +70%

05

Products look alike

Agents build interfaces from the same templates. Nobody thinks about how easy the product is to use.

WHAT OPEN SOURCES SHOW

Uber

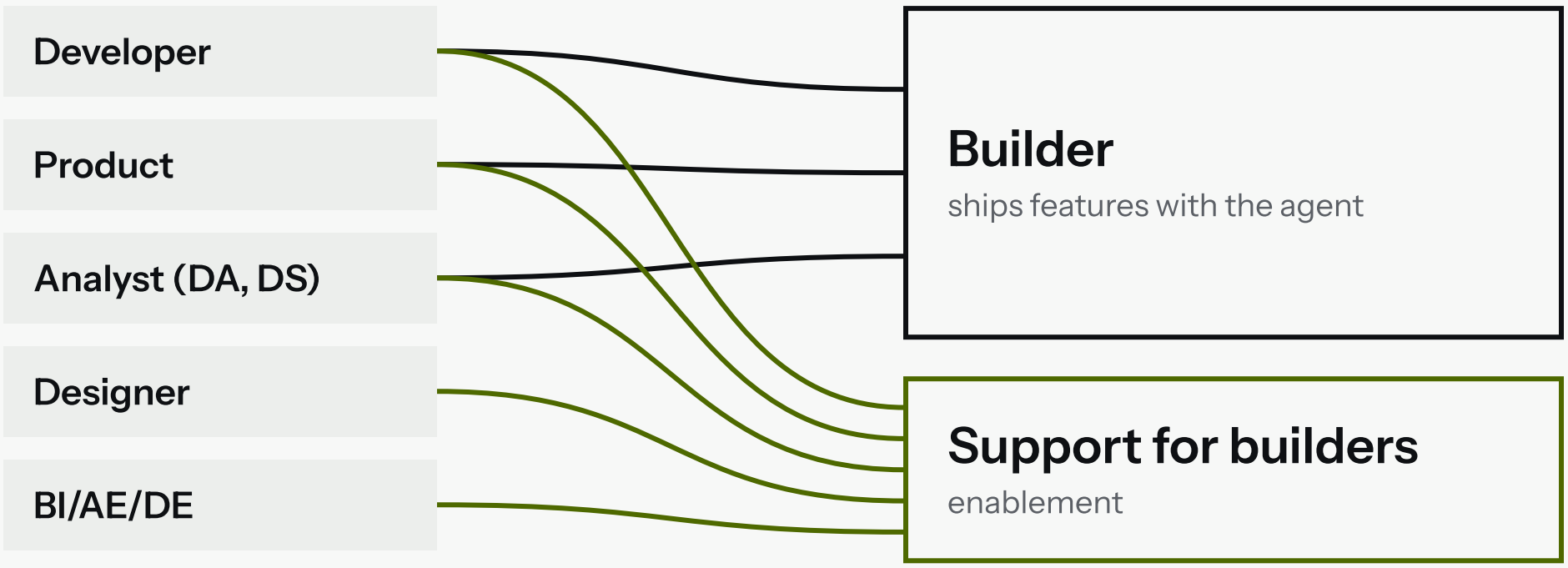
teams cling to the first prototype and take it for the finished solution

Intercom

taught agents its own set of interface elements: code using it went from 7% to 96%, yet a quarter of screens still have errors

How roles change.

My generalised scenario from the pilots. Some developers, product managers and analysts become builders; every role becomes their support.



WHAT CHANGES IN EACH ROLE

Developer	~~Writing code~~ → Tasks for the agent, checking its code. Plus part of the product and analyst jobs
Product	~~Mock-ups, simple features~~ → What to build and in what order, customers. Plus part of the developer and analyst jobs
Analyst (DA, DS)	~~SQL, routine pulls, simple A/B tests~~ → Complex tests and research, teaching other roles their skills
BI/AE/DE	~~Metrics, data marts, dashboards, pipelines from templates~~ → Running the domain knowledge base and shared metrics for the agent (semantic layer), checking its answers (evals), owning which data is right, keeping an eye on what builders make

Who becomes a builder.

WHO IS EASIER TO RETRAIN

> My observation: product managers and analysts. They learn to code with an agent quickly. Thinking about the business and making decisions are harder to teach.

OTHER VIEWS

- Intercom: the best engineers will take on part of the product job
- Atlassian: only a few product managers and designers will become builders
- Harvard and P&G: with AI, people from R&D and from commercial propose more balanced solutions; without it each pulls their own way

WHAT HAPPENS TO NEWCOMERS

⅓ → <⅕

share of newcomers in engineering, Bain, 155 companies

>6x

the top 5% of engineers do more than the median, Intercom

What to measure if you run a pilot.

01

Artefacts: is there an effect

What builders made — features, tests, reports. Did it reach users and move a metric

02

Logs: are there attempts

Who tries to do the task themselves with the agent, how often, and at which step they give up and go to a specialist

03

Barriers: what fails in the agents

Why the job someone came to the agent for (JTBD) was not done: no data or access, a wrong answer, a missing skill

Sources and the full piece: stop-token.com/blog/product-builders