Analysis Report for Assessing Developer AI Usage

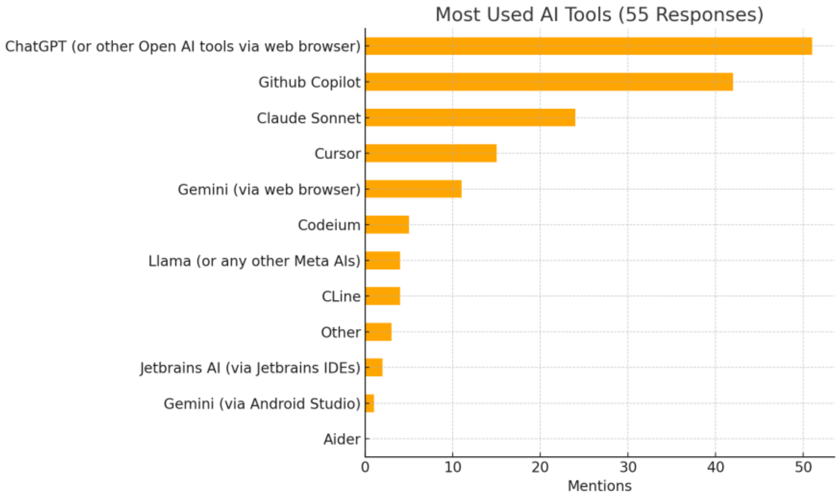
(Compiled as part of an internal company-wide survey conducted across the engineering teams)

Ashutosh Verma | Shuru Labs | June 10, 2025

Synopsis

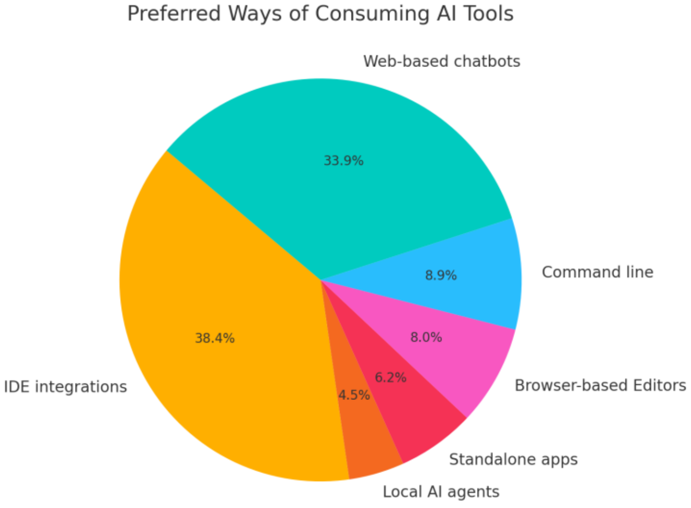
This report summarizes the results of an extensive survey prepared to gather data on developer AI usage patterns across the company. Responses from a total of 55 engineers were collated and analyzed over a two-week period.

TOOLS



Developer usage of the GPT family leads Claude, Gemini, etc., which form a strong secondary tier. Other (currently ongoing) studies will shed further light on the implications, causes, etc., behind these patterns – as well as help filter out false positives & corrections required for sampling problems.

MODALITIES

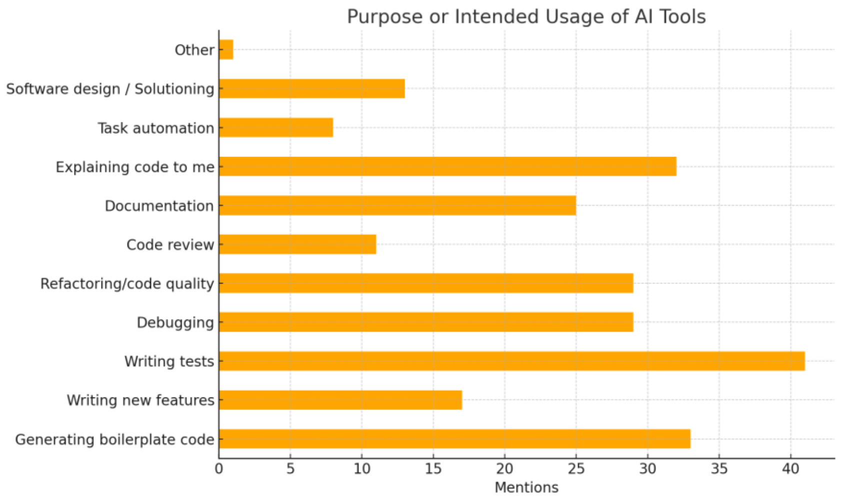


1. AI tool consumption using web-based tools is slightly higher than IDE integrations. CLI and local agent usage are also statistically significant outliers.

2. It could hence be *implied*, that developers with a workflow suited for automation, prefer IDE integrations, since web based chatbots must be manually seeded with context.

3. While modalities weren't tried to models yet, a ready access of GPT based plugins may have skewed the model distribution graph towards the GPT side.

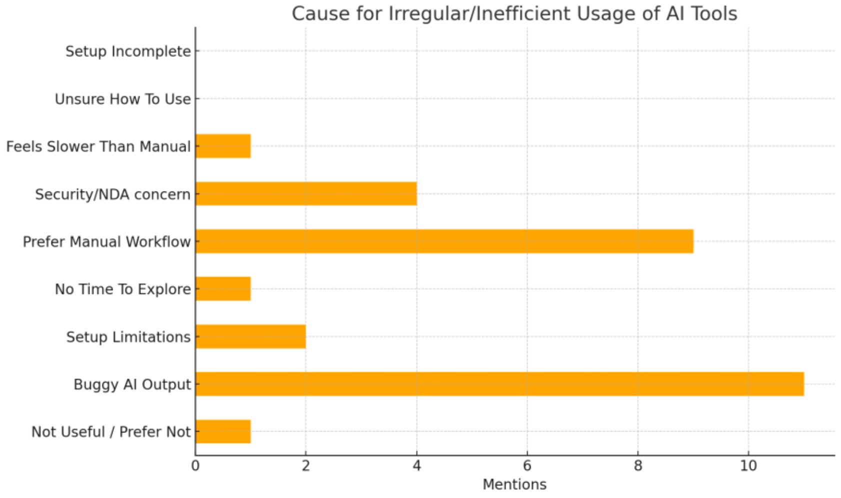
PURPOSE



The intended purpose-of-use graph is quiet spread-out & varied, with the most common one being AI tool usage for writing test cases, followed closely by several use-cases such as: code snippet explanation, boilerplate generation, refactoring, debugging, and documentation.

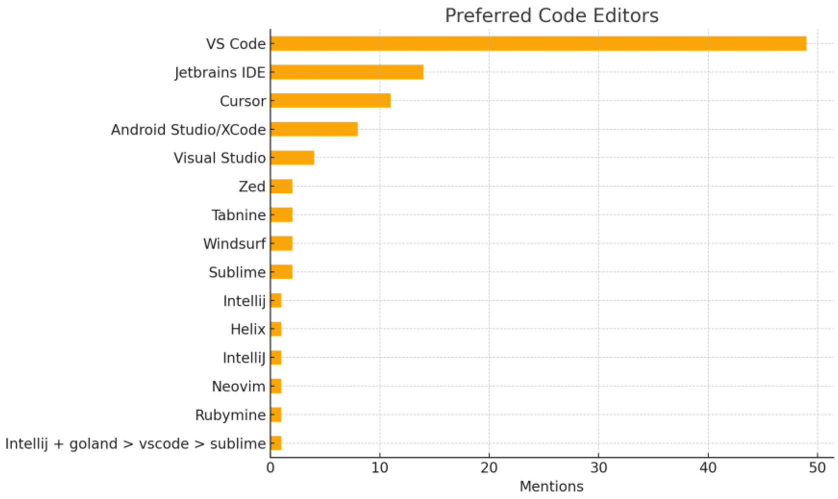
Since the intended purpose-of-use will vary as per individual responsibilities (viz., a TL/EM may use AI tools differently than an IC), further study is warranted to explore role-specific patterns.

HINDRANCES



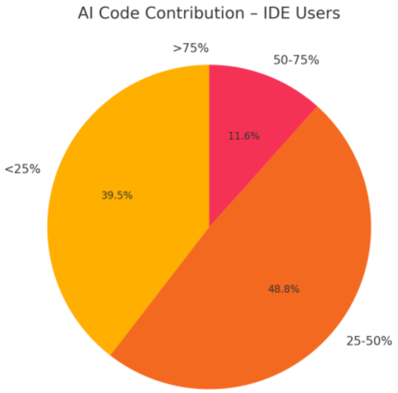
While most are not opposed to AI usage, buggy AI output, a preference for a manual workflow, and security/client NDA concerns, happen to be the top blockers for AI tool adoption.

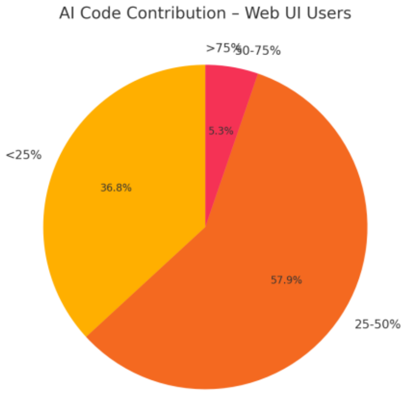
CODE EDITORS



VS Code clearly dominates the development environment, followed by JetBrains IDEs and Cursor. This indicates a strong compatibility for IDE-based AI integrations. There is also a potential for further studies which would track and analyze IDE extensions usage.

PERCENTAGE OF AI GENERATED CODE





AI-generated code contribution is significantly low for both IDE and web UI users, with IDE users leaning more toward >50% generation. This split justifies dual-mode instrumentation for future experiments.

Conclusion.

The survey confirms both the readiness and feasibility for Shuru-wide AI experiments aimed at improving and increasing structured AI tooling adoption among developers. While the survey was only engineering-focused this time, product managers can also be included in future iterations.