

AI + Engineering = Magic at Airbnb

Developer Productivity Engineering Summit
San Francisco 9/20/23



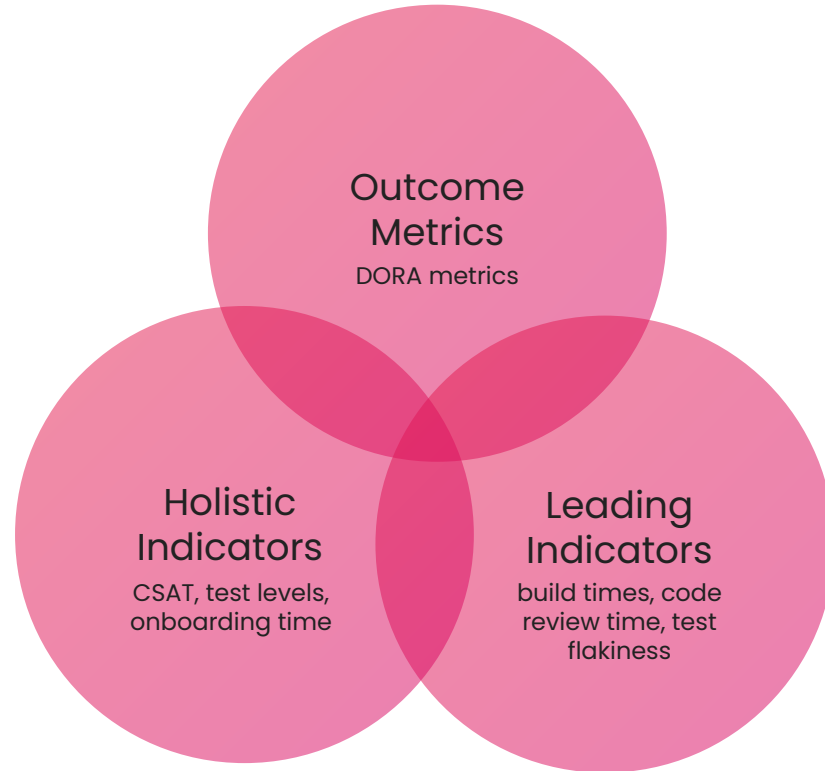
Szczepan Faber

Engineer, Developer Platform, Airbnb

Please list any improvements or product launches (in the last 6 months) that have positively impacted your developer workflow

Airbnb's DevX Survey

Holistic DevX

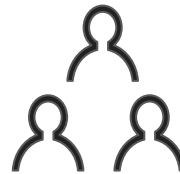




ChatGPT



GitHub Copilot



Build your own

ChatGPT

How many years were there between the release of GPT-3 and GPT-4?

- A. 1
- B. 2
- C. 3
- D. 4

ChatGPT Use Cases

~30% monthly PR authors
use Airbnb's One Chat
(Jul/Sept '23)

Coding

Writing Tests

Refactoring

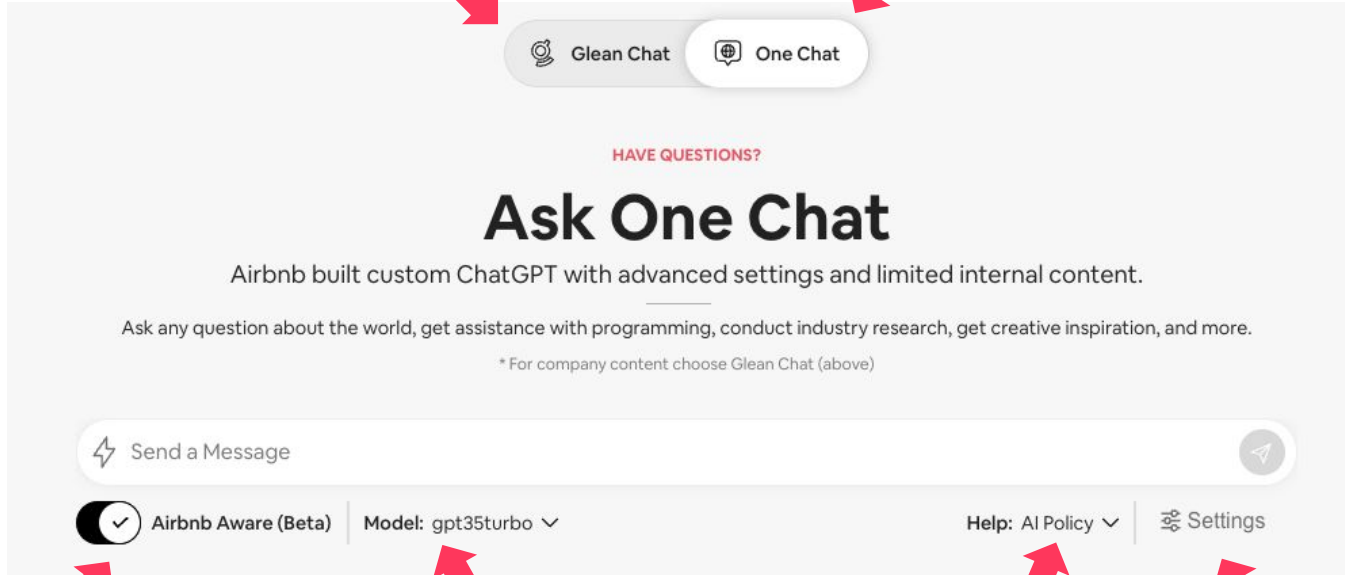
Documentation

Debugging

Support

Use Glean Chat for 100% Airbnb specific questions: workflows, policy documents, meeting notes, and more.

Use One Chat for everything else, including code!



Internal knowledge base aware (retrieval augmented generation - RAG)

Pick your model, up to GPT-4 32k

Airbnb's AI policy

Tune settings: system message, "temperature", token limit

Only use internal chat

Otherwise things can go wrong...

Blocked in some countries over privacy concerns

Large enterprise bans ChatGPT after sensitive code leak

AI copyright complaints

GPT Input

Logic in system message,
context + data in user message

Anatomy of GPT input

One system message, then interleaved user and context

- System message: frame what the model “knows” and “cares about”
 - Who is the assistant
 - What do they help their users with?
 - What do they know
 - ... about the user *specifically*
 - ... about their platform *generally*
- User message
 - What help do you need? (be specific!)
 - You can attach context, trick the model into knowing things

Airbnb's One Chat

maxTokens
3000

The maximum number of tokens to generate in the chat completion. The total length of input tokens and generated tokens is limited by the model's context length.

Temperature
0.8

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

stop

String where the API will stop generating further tokens.

System Message

You are an AI assistant for Airbnb employees. Help make them productive with details, helpful information, and tips.

Default system prompt



****bold**** *_italics_* ~~~~strike~~~~ [link](https://)

How many tokens are in the Declaration of Independence?

- A. 1.6k
- B. 5.1k
- C. 11.4k
- D. 22.5k

18k Tokens in *Bartleby, the Scrivener* (Herman Melville)

\$1.12 How much it would cost to run that through GPT-4 32k

27k Tokens in *The Little Prince* (Antoine de Saint-Exupéry)

\$1.30 Cost to run that through GPT-4 32k

~40–90% Amount saved by adding “be concise” to prompt

1.3 Average tokens per word

~50 Cost ratio GPT-4 vs GPT-3.5 Turbo

~\$1M Cost to train 13 billion parameter model on 1.4 trillion tokens

Source: <https://github.com/ray-project/llm-numbers>

Retrieval Augmented Generation (RAG)

How Airbnb's One Chat is "Airbnb aware"

Enter RAG (Retrieval Augmented Generation)

Tell the model what it needs to know

- Find relevant context:
 - Things a user recently did (docs, code?)
 - Articles, code, etc. that are related
- Combine the context into user or system prompt
- “Relevance” can be achieved in many ways:
 - If you already have search + preview system, reuse it
 - Embedding-based retrieval: use LLMs to find “relevant” snippets
 - Whatever happened recently in the system
 - Code / docs nearby in the filesystem

What is Onebrain?

Before RAG

Onebrain is an AI-driven software solution developed by Airbnb that aims to improve efficiency, productivity, and decision-making across the company. It integrates data from various sources and uses machine learning algorithms to analyze and make sense of the data. This helps Airbnb employees make better decisions and streamline processes by providing insights, predictions, and recommendations based on the analyzed data.

After RAG

You are an assistant for Airbnb employees. You know a series of facts about Airbnb:

{output of one.airbnb.com search}

What is Onebrain?

OneBrain is a Python package and command line interface (CLI) developed by Airbnb. It is a versatile tool designed to power a scientific project ecosystem, providing integration with Redspot and other functionalities. OneBrain serves as a comprehensive platform for managing and executing various tasks within the Airbnb development environment. It helps Airbnb employees streamline their workflows, automate processes, and collaborate more effectively on projects.

Generate iOS project

Without Airbnb Aware

1:50:14 PM
How to generate iOS project?

1:50:38 PM

To generate an iOS project, you would typically use Xcode, Apple's integrated development environment (IDE). Here are the steps:

1. Install Xcode: You can download it for free from the Mac App Store.
2. Open Xcode: After installing, open Xcode from your Applications folder.
3. Create a new Xcode project: Click on "Create a new Xcode project" (or choose File > New > Project).
4. Choose a template for your new project: When the dialog appears, select iOS at the top, then choose one of the template (e.g. Single View App) and click on "Next".
5. Configure your project: In the following dialog, enter your project details — Product Name, Team, Organization Name, and Organization Identifier. Make sure Swift is selected as the language, and UIKit as the User Interface option. Click "Next" when you're done.
6. Choose a location to save your project: Specify a location where you want your new project to be saved and click "Create".
7. And voila! You now have a new iOS project started.

Remember to regularly build your project (CMD + B) to make sure it compiles without errors, and run your project (CMD + R) to test it in the simulator.

Send a Message

X Airbnb Aware (BETA) Settings

RAG switch



Generate iOS project

With Airbnb Aware



1:51:02 PM

How to generate iOS project?



1:51:25 PM

To generate an iOS project for prototyping purposes, you can follow the instructions explained in the "Getting Started on iOS" documentation:

1. To start, use the command "`./bear make:module`". This opens up a prompt that would aid in the creation of a module.
2. Select the demo module type, and follow the prompts to create a new view controller and generate a dev app for your new module.
3. Once you've finished, you should now have an Xcode project that launches directly into your demo module's view controller.

The generated Xcode project should now be ready to iterate on and modify according to your feature requirements. An example of a prototype demo module can be found in the [HostDLSDemo](#).

For further details and guidance, the documentation can be found [here](#).

RAG switch



Send a Message



Airbnb Aware (BETA)

Settings

Copilot

Game changer?

GitHub about Copilot

Improving developer satisfaction & Conserving mental energy

74% Focus on more satisfying work

88% Feel more productive

96% Faster with more repetitive tasks

Development speed

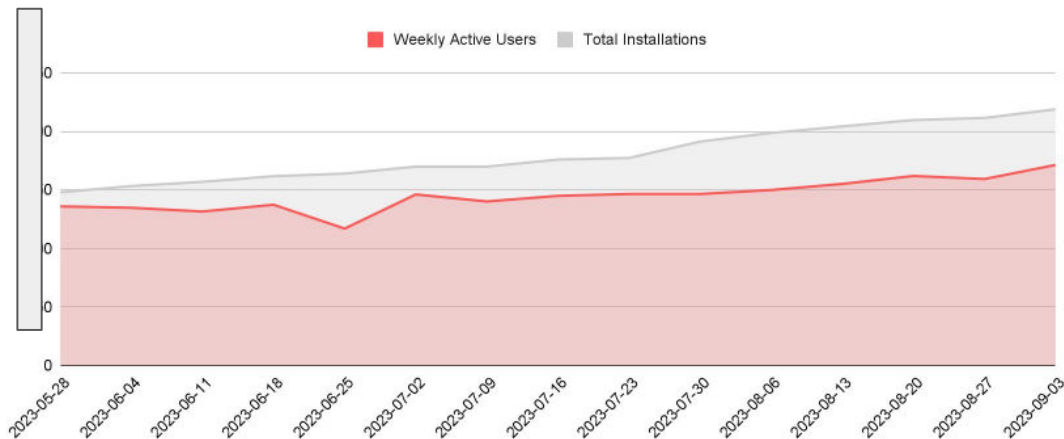
95 Developers given the same task

55% Less time to finish

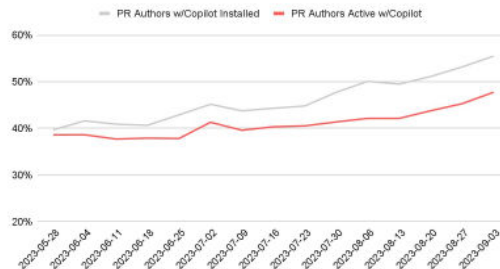
Copilot

- 55% of weekly PR authors have Copilot installed. 48% actively using*
- 58% of PRs are from authors with Copilot installed

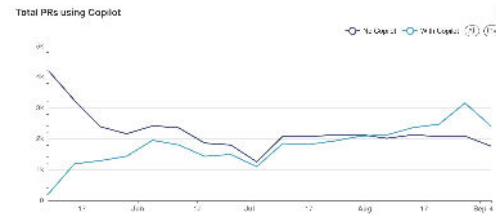
Total Installations & Weekly Active Users



% of PR Authors with Copilot



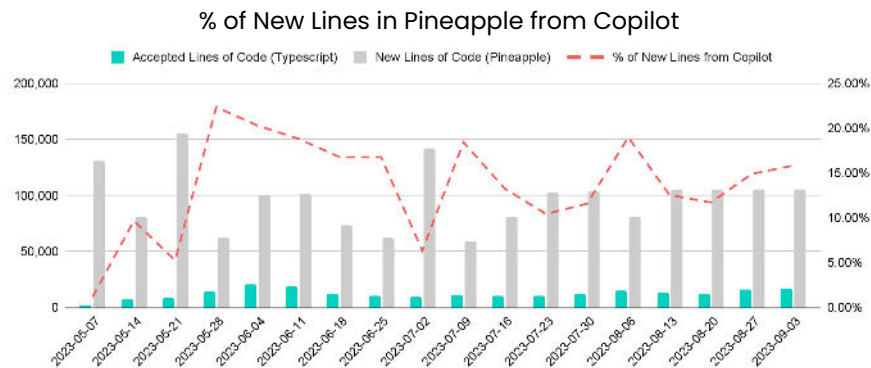
Total PRs from Authors with Copilot



* Active users as reported by GitHub

Copilot Engagement VS Code Only

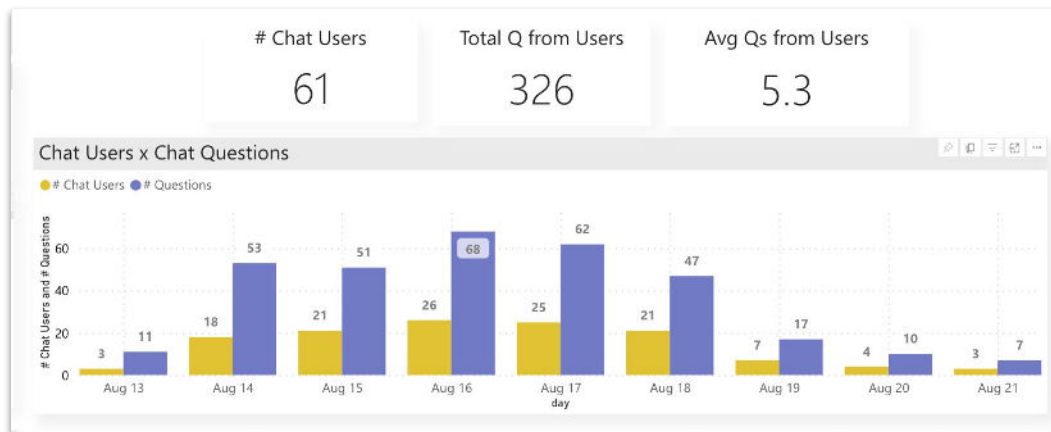
- 13k weekly accepted suggestions
- 27% accept rate. 2.5 lines per suggestion
- 16% of new lines of code for Web from Copilot*



* % of new lines from Copilot estimated by accepted lines of typescript / new lines of code committed to Pineapple repo. Since VS Code is used for more than just Pineapple this is likely overcounting.

Copilot *Chat*

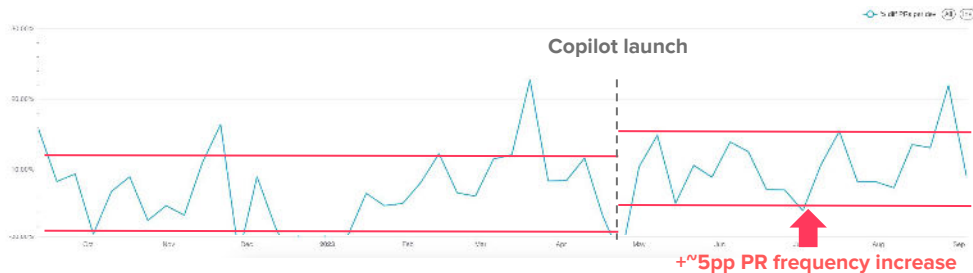
- Copilot X feature
- Only for VS Code for now
- ~60 WAUs. 5.3 questions per user.
- Mixed feedback. Some rave. Others say slow, not as helpful as ChatGPT.



Impact on Productivity

- Copilot developers submit PRs at ~7-15% higher frequency than non-Copilot developers (up ~5pp since Copilot launch)

PR frequency of developers with Copilot vs without (% difference)



Outside research

- Copilot users [completed a task 56% faster](#) than those without. 35% self-estimated increase in productivity.
- [97% of developers](#) said Copilot made them more efficient and productive.

Build Your Own



Innovate beyond Chat and Copilot

Azure OpenAI Egress Proxy

Give me the APIs

- [API-compatible with Azure OpenAI REST APIs](#)
- Handles:
 - Authentication
 - Load balancing across multiple Azure OpenAI instances
 - Usage audit trail
- Humans authenticate with @airbnb.com account
- Services interact via standard mesh service URL
- Supporting Airbnb Chat + many hackathon projects

Internal Library (Python example)

```
from airbnb_identity import DefaultAirbnbCredential
import openai

openai.api_base = "https: [REDACTED]"

auth = DefaultAirbnbCredential().authenticate(openai.api_base)

openai.api_key = auth.headers["Authorization"].split()[-1]
openai.api_type = "azure_ad"
openai.api_version = "2023-03-15-preview"

response = openai.ChatCompletion.create(
    engine="gpt4",
    messages=[
        {
            "role": "system",
            "content": "Assistant is a large language model trained by OpenAI.",
        },
        {"role": "user", "content": "Who were the founders of Microsoft?"},
    ],
)

print(response["choices"][0]["message"]["content"])
```

OpenAI SDK

Authenticate

API version

Sandcastle - prototyping framework

- Encourage prototyping culture!
- Fully customizable frontend and backend
 - Python (Streamlit)
 - JavaScript (React)
 - R (Shiny)
- Data connectors
 - Data warehouse (Presto, etc)
 - S3 buckets
 - Internal & production APIs (Superset, etc)
 - Google sheets
- Permanent production and testing urls
- Managed access control, secured by Google IAP and AirMesh

Review GPT

What if AI reviewed code?

Tag AI reviewer



ChatGPT review



Conversation 2 Commits 1 Checks 2 Files changed 1

commented 3 hours ago · edited ·

Change summary:

Prewarm simulator to avoid xcodebuild timeouts when trying to launch the simulator.

How was it tested?

htl [redacted] 113

Please review:

@svc-review-gpt

Prewarm simulator 733e8eb

deployboard commented 3 hours ago

ReviewGPT AI-Generated Code Review

1. While this PR seems to make sense conceptually, it's also worth considering that Apple's Simulator app may not be running by the opening of the Simulator before the tests since killing the Simulator processes is part of the test workflow. The opening of the simulator might be ignored, or the workspace may have some unexpected state when the tests run. Moreover, the `open -b com.apple.iphonesimulator || true` command runs asynchronously, and there is no certainty the simulator app will be ready when the tests start running.

I propose a more robust way handling simulator prewarming by utilizing `simctl` command which provides direct capabilities for interacting with iOS simulators. For example, you can directly boot a simulator device using `xcrun simctl boot <device>`, where `<device>` is the id of the simulator, and you can ensure this operation completes before running the tests.

Here is my recommendation:

```
system_and_log 'pgrep Simulator | xargs kill || true'
```

```
# Prewarm simulator to avoid xcodebuild timeouts when trying to launch the simulator.
-system_and_log 'open -b com.apple.iphonesimulator || true'
+system_and_log 'xcrun simctl list devices | grep "Shut Down" | cut -d "(" -f2 | cut -d ")" -f1 | xargs -I {}
```

This command lists all devices, filters the ones that are shut down, extracts the id, and boots them.

Note: This command attempts to boot all shut down simulators. If you want to boot a specific simulator, you would need to identify it by its identifier instead of the `grep "Shut Down"` command.

Review GPT


User prompt

```
# It's important that any dynamic input sources are truncated here (such as git diff output)
# otherwise, the API response will fail if the number of tokens exceeds the allowed threshold.
prompt_text = "\n".join([
    f"PR title:``{pr_title}``",
    f"PR description:``{pr_description}``",
    f"Provide useful suggestions based on the lines in the following git diff:``{diff_text}``",
])
```

Triple quotes



Check out
copilot-explorer



System prompt

```
def system_prompt():
    return "\n".join([
        f"Assistant is {REVIEW_GPT_GITHUB_USER} aka \"ReviewGPT\", an AI that does code reviews of Airbnb-internal code in Github.",
        "Assistant will review the git diff, where lines starting with '-' are being removed, lines starting with '+' are being added, and other lines are surrounding context.",
        "Assistant does not introduce itself. Assistant's responses ONLY include the actual code review without any other context.",
        "Assistant ONLY uses backticks `` to surround code snippets and never uses backticks to format non-code.",
        "Assistant does not criticize for minor style issues and does not criticize for things that are not in the diff, like missing PR description items.",
        "Assistant offers concrete, actionable, and numbered suggestions based on best practices and observed bugs, and does not suggest tangential, vague, nitpicky, or out-of-scope changes.",
        "If Assistant doesn't have any constructive review comments to add, it will simply respond 'LGTM'.",
        "100% of the response is always in github markdown (without any other text or headings, and without escaping the markdown), suitable for copying into a github comment directly.",
    ])
```

Review GPT

Cost management

This is per-token price



```
# Using GPT-4 32k is 2x the cost of GPT-4 8k.
# Here we pre-compute the tokens and use the smallest possible model to reduce cost.
# Note that this approach of estimating tokens is not guaranteed to be accurate,
# but during testing, the number of tokens was very close (within 10 tokens).
prompt_to_str = "\n".join([message["content"] for message in prompt_messages])
estimated_token_count = tokenizer.token_count(prompt_to_str)

if (estimated_token_count > OPENAI_API_ENGINE_LARGE_TOKEN_THRESHOLD):
    engine = OPENAI_API_ENGINE_LARGE
else:
    engine = OPENAI_API_ENGINE_SMALL
```

“For certain tasks, ChatGPT has made me 10x more productive” –
engineer at Airbnb

“Then the day after chatgpt came to Airbnb, I had the ability now to build all of the systems I asked for for the last 4 years myself, in a fraction of the time.” – engineer at Airbnb

“ChatGPT allowed our team to deliver results that previously appeared operationally impossible” –
eng leader at Airbnb

“Chat is unbelievably good. I have been enjoying asking really detailed questions about processes that product teams have where I have access to the documentation over like 9.5 years.” – employee at Airbnb

ChatGPT

GitHub Copilot

Build your own

Thank you!

Questions?

