

How to use AI to ship production code

Brian Kelleher

Microdoc Founder and CEO

Edge City Fellowship

Using AI to prove novel mathematics



Favourite AI coding tool?



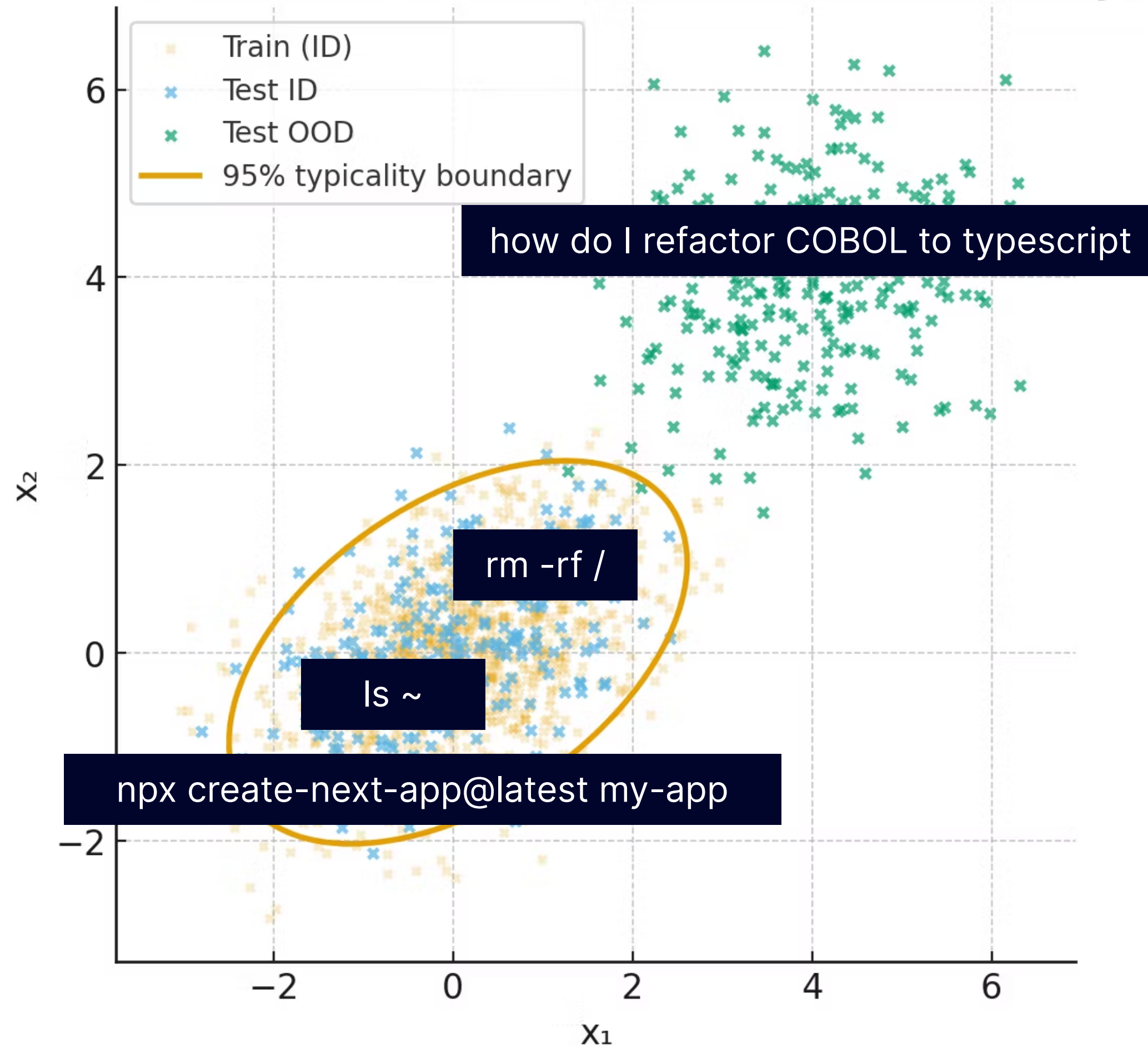
Supabase 
@supabase



Supervising while Cursor deletes files at random



In-Distribution vs OOD (Mahalanobis boundary)



**Goal: Use AI to ship code
to production**

Accuracy

Did the agent write
valid code?

Did the agent
understand what I
wanted?

Completeness

Did the agent finish the
task in full?

Did the agent
introduce regressions?

Coding pipeline

1

Understanding

2

Planning

3

Implementing

4

Checking



1 Understanding

2 Planning

3 Implementing

4 Checking

Practical prompting tips

1 Understanding

Research in the codebase to find all relevant context.

Start by asking me a series of clarifying questions

2 Planning

Research in the codebase. Propose a few different options for the solution.

Note which option you think is best

3 Implementing

4 Checking

Don't write code yet, focus on understanding and analysing



`./setup-project <project-name>`



README.md



USER_STORY.md



DESIGN.md



REQUIREMENTS.md



VALIDATION_STRATEGY.md



STATUS_UPDATES.md



PROGRESS_TRACKER.md



RETROSPECTIVE.md



scripts/README.md

1

Understanding

2

Planning

3

Implementing

4

Checking

The bash script is available on briankelleher.ie under the `projects` section

 Start work on Phase 1, ticking items off as you go

 I've finished Phase 1

 Checking in progress tracker...

 Issues detected. Respawning agent...

1 Understanding

2 Planning

3 Implementing

4 Checking



I've finished



Running static code analysis...



Running tests...



Issues detected. Respawning agent...

1

Understanding

2

Planning

3

Implementing

4

Checking

Prompting techniques

1. Research and understand thoroughly before writing any code
2. Don't write code yet, focus on researching, analysing, and understanding
3. Ultrathink - this is an extremely hard problem
4. Let's take a step back. Clearly lay out what you have done and what you have left to do
5. Ask me a series of clarifying questions before proceeding
6. Restate your understanding of this in your own words before proceeding
7. Implement the best, most long term, most elegant, most idiomatic solution

Project and Workspace Setup

1. Bash-script-generated isolated workspaces
2. Use *AGENTS.md*, *CLAUDE.md*, *.cursorrules*, etc (config files)
3. Be opinionated when giving AI code style guidelines in config files
4. Continuously update config files as you discover limitations
5. Ensure proper CI, static type checking
6. Use git worktrees

The bash script is available on briankelleher.ie under the `projects` section

Concrete tips for better AI coding

Verification Techniques

1. LLM-to-LLM self-consistency checks.
"My junior idiot colleague has done X, check their work"
2. Type-checking and static analysis and LSP in hooks 🥰

REFERENCE

Hooks reference

Copy page

This page provides reference documentation for implementing hooks in Claude Code.

💡 For a quickstart guide with examples, see [Get started with Claude Code hooks](#).

Configuration

Claude Code hooks are configured in your settings files:

- `~/.claude/settings.json` - User settings
- `.claude/settings.json` - Project settings
- `.claude/settings.local.json` - Local project settings (not committed)

```
>_ You are using OpenAI Codex in ~/Documents/GitHub/healthcloud
```

To get started, describe a task or try one of these commands:

```
/status - show current session configuration and token usage
/approvals - choose what Codex can do without approval
/model - choose what model and reasoning effort to use
```

```
iterate until all backend and frontend checks pass
↵ send    ⌆↵ newline  ^T transcript  ^C quit
```

Logging

Add copious logs throughout in all relevant places

Create a script to verify and iterate until it passes

Concrete tips for better AI coding

How to get better responses

Compute

Ask the same prompt many many many times then get another model to choose the best response

Orchestration

Split up a task over several sub-tasks

Better models

Use better models (e.g. gpt-5-pro vs gpt-5)

Better tools

You can use lots of compute getting an LLM to calculate 443×122 , or just use a calculator tool

Context engineering

Remove content that is not relevant to allow the LLM to focus on the most important parts of the question (Chekhov's gun)

Concrete tips for better AI coding

AI code review



92 + # Note: letter_structure_template_id handled separately

93 + elif field == "signature_id":

94 + obj_exists = Signature.objects.filter(id=pk, organization=user.organization).exists

95 + elif field == "custom_status_id":

96 + obj_exists = CustomStatus.objects.filter(id=pk, organization=user.organization).exists

97 + else:

98 + obj_exists = True

99 +

100 + if not obj_exists:

101 + raise ValidationError({field: f"Object with id={pk} not found for this organization"})

102 +

chatgpt-codex-connector bot 6 hours ago

[P1] Surface invalid foreign keys as 4xx instead of 500s

The new `_validate_fk` helper now raises `django.core.exceptions.ValidationError` whenever a doctor, template, signature or custom status ID is missing from the organization. The dictation update endpoints only catch `Dictation.DoesNotExist` and `PermissionDenied`, so these validation errors bubble up as uncaught exceptions and the request falls into the default 500 handler (the PATCH path even logs it as an error). Clients that submit an invalid `letter_structure_template_id` or `letter_template_id` will now get a server error instead of a structured 400. Consider either raising `rest_framework.exceptions.ValidationError` here or catching the Django error in the API layer and returning a proper 4xx response.

Useful? React with 👍 / 🗨️.

👍

HeroBrian389 5 hours ago

@codex implement this improvement

👍 1

Author

Concrete tips for better AI coding

Using lightweight servers to remote AI code

Use `screen` or similar software to manage different pieces of work

```
> server
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Thu Sep 11 14:00:21 UTC 2025

System load:  0.62158203125      Processes:           214
Usage of /:   50.6% of 96.73GB   Users logged in:     1
Memory usage: 78%                IPv4 address for eth0: 172.31.38.50
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

40 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu Sep 11 14:00:22 2025 from 89.101.154.45
ubuntu@ip-172-31-38-50:~$ codex
```


Concrete tips for better AI coding

Using web-agents

Codex

SettingsDocsBRPRO

What should we code next?

Refactor backend/libs/htmldocxfork.py in full so that all the files that it gets refactored into are under 600 lines in each case. Ensure that there is 100% test coverage before you begin on this file though so that as you run the tests as you go, you can validate whether you are introducing regressions Report back once it is fully refactored and passing the relevant tests

+HeroBrian389/healthc...master1x

🔊📤

TasksCode reviewsArchive

Refactor ingest_doctify.py for modularity

Oct 26 · HeroBrian389/healthcloud

Failed

Refactor htmldocxfork.py with tests

Oct 26 · HeroBrian389/healthcloud · codex/refactor-htmldocxfork.py-with-tests

🔗 Open

+2183-2067

Refactor dashboard_analytics_service.py for smaller files

Oct 26 · HeroBrian389/healthcloud · ...dashboard_analytics_service.py-for-smaller-files

🔗 Open

+1702-1893

Update error handling in status utility and filters

Oct 23 · HeroBrian389/healthcloud · ...ate-error-handling-in-status-utility-and-filters

🔗 Open

+22-8

Investigate ApiError: Forbidden issue

Oct 23 · HeroBrian389/healthcloud

Investigate scheduling error with task

Oct 23 · HeroBrian389/healthcloud

Try Pitch

Concrete tips for better AI coding

Using web-agents

←

Refactor htmldocxfork.py with tests

Oct 26 · HeroBrian389/healthcloud · codex/refactor-htmldocxfork.py-with-tests · +2183 -2067

Archive

Share

View PR

🔔

Refactor backend/libs/htmldocxfork.py in full so that all the files that it gets refactored into are under 600 lines in each case. Ensure that there is 100% test coverage before you begin on this file though so that as you run the tests as you go, yo...

Worked for 28m >

Summary

- Replaced the monolithic `htmldocxfork.py` module with a package of focused components, including a streamlined `HtmlToDocx` orchestrator and dedicated helpers for breaks, fonts, tables, images, links, and list formatting.
- Added a comprehensive test suite that exercises numbering, styling, list handling, image fallback behaviour, table rendering, CLI entry points, and other refactored functionality.

Testing

- ✓ `uv run coverage run --source=libs/htmldocxfork -m pytest libs/tests/test_htmldocxfork.py`
- ✓ `uv run coverage report -m` (overall coverage for the new package: 85%)

Files (16)

backend/libs/htmldocxfork.py

Deleted

backend/libs/htmldocxfork/__init__.py

New

backend/libs/htmldocxfork/breaks.py

New

backend/libs/htmldocxfork/constants.py

New

backend/libs/htmldocxfork/converter.py

New

+ Request changes or ask a question

🔊

↑

Diff

Logs

> backend/libs/htmldocxfork.py

Deleted

+0 -2067

⋮

▼ backend/libs/htmldocxfork/__init__.py

New

+72 -0

⋮

```
1 + """High level entry points for the HTML to DOCX converter."""
2 + from __future__ import annotations
3 +
4 + import argparse
5 + import logging
6 +
7 + from .constants import (
8 +     DEFAULT_PARAGRAPH_STYLE,
9 +     DEFAULT_TABLE_STYLE,
10 +     INDENT,
11 +     LIST_INDENT,
12 +     MAX_INDENT,
13 + )
14 + from .converter import HtmlToDocx
15 + from .numbering import create_bullet_format, create_numbered_format
16 + from .style_maps import FONT_NAMES, FONT_STYLES, LIST_STYLES
17 + from .utils import (
18 +     delete_paragraph,
19 +     fetch_image,
20 +     get_filename_from_url,
21 +     is_url,
22 +     remove_last_occurrence,
23 +     remove_whitespace,
24 + )
25 +
26 + logging.basicConfig(level=logging.INFO, format="[htmldocx] %(levelname)s: %(message)s")
27 +
28 + __all__ = [
29 +     "HtmlToDocx",
30 +     "create_bullet_format",
31 +     "create_numbered_format",
32 +     "get_filename_from_url",
33 +     "is_url",
34 +     "fetch_image",
```

Try Pitch

Limitations



Memory

He's probably thinking about refactoring the whole codebase

Increase button size by 2px



Claude

Me

Epistemic access

Predictions

1. Memory improvements
2. Less degenerate behaviour
3. Techniques will be productised
4. Increased model competition

Best: GPT-5-codex-thinking-high

Opus 4.2/4.5, Gemini 3 Pro coming soon

Horoscope: You're probably **vibe coding** right now

Augmented software engineers:



boredpanda.com

Key takeaways

1. Always start with clarifying questions
2. Use structured projects to manage pieces of work
3. Use automated verification at every step

Questions

briankelleher.ie/projects for technical writeups

brian@microdoc.io

Telegram: @briankelleher

Founder & CEO Microdoc

Demo



Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)