



TRAINING

Hands-on AI-Assisted Development

IT track Day 1

Supplement: The limits of AI and how to work with them



Do not redistribute these materials

Redistribution of these materials is prohibited.

These materials are copyrighted by Givery, Inc. Use them only for your own learning and work reference. Do not share them outside your company, on social media or with third parties.

During the course, do not take screenshots or recordings that show these materials.

What AI does well and what it does poorly

AI is strong at writing, but **does not guarantee correctness**

	Good at	Poor at
1	Generating boilerplate: input forms, lists and other tedious code.	It does not know client specs or company rules unless you give them.
2	Explaining existing code and adding comments. Less reading for you.	It knows library specs only up to the point its training ended.
3	Unifying names and types, even across dozens of places.	It confidently writes methods that do not exist.

Tricks do not remove the weaknesses. They come from how AI works, so take them as given.

Errors written with confidence

You cannot tell if it is right **until you run it**

Here is the kind of error you get when you ask for total hours in the Worklog app. It reads well, but the method on line 2 does not exist.

```
// A form you may get back (it does not run as is)
const total = records.sumBy('hours');
// Arrays have no sumBy. Running it stops with a TypeError.

// A form that actually works
const total = records.reduce((sum, r) => sum + r.hours, 0);
```

Reading alone, this line looks fine. Open it in the browser and Console shows the error at once.

What a model can read at once **has a limit**

Early instructions drop out

A model can read only a fixed amount at once. As the conversation grows, the oldest parts drop out and the rules you set first stop working

The symptoms take 3 forms

Code you just fixed goes back to how it was. Naming rules are ignored. It starts from premises that do not exist. All of these increase in long conversations

The fix: open a new chat instead of continuing

Pushing on in one thread makes it worse. A new chat with only the needed context is faster

When you give the same context every time, put it in a file: the Session 1 afternoon's instructions file

The newer it is, the more it gets wrong

It does not know what came after its training

Each model has a training cutoff. Library specs and changes released after it are not included

Instead of saying it does not know, it answers the old way

It confidently writes removed functions and old syntax. You notice only when you run it

Give it official information, then ask

Ask with the library and version written in. Ask with the relevant page of the official docs open. These 2 steps cut errors a lot

Remember that AI is weaker on newer technology, and you will decide faster

When the output is too much

Do not accept more than you can read

1

Narrow each request to what you can read

Ask for one file or one function at a time. Building it all at once leaves code you cannot check

2

Get a diff and read only the changes

Do not ask for the full text; ask what changed and how. You read a tenth as much

3

Decide what to check before you ask

First decide what output means it is right. Otherwise you end up reading everything

4

What you pass without reading is still your responsibility

"AI wrote it" is no excuse. The last person to touch it decides if it is ready to ship

If you cannot keep up with checking, you are usually asking for too much at once



Givery, Inc.

Teito Shibuya Bldg. 8F, 15-13 Nampeidai-cho, Shibuya-ku, Tokyo 150-0036, Japan