AI Agents for Daily Life Tasks

Victor Dibia

I have been building AI agents for three years.
I have come to believe they should not be just for work, but for life.

The principles that apply at work translate directly to life tasks.

Non scholae, sed vitae discimus

Not for school, but for life.

My high school English teacher. What we learned in school was meant to be applied outside it, not just to pass exams.

Most agent use is work

- Deep research
- Writing code
- Slide decks

Beyond that, not much.

I think this leaves value on the table.

About me

Victor Dibia · Principal Engineer, Agentic Systems & Data

- Previously Microsoft Core AI and Microsoft Research
- Core developer for AutoGen, 52K stars
- Creator of AutoGen Studio and Agent Optimizer in Foundry Agent Service
- Author, *Designing Multi-Agent Systems* · PhD · IEEE Senior Member

What you’ll take away

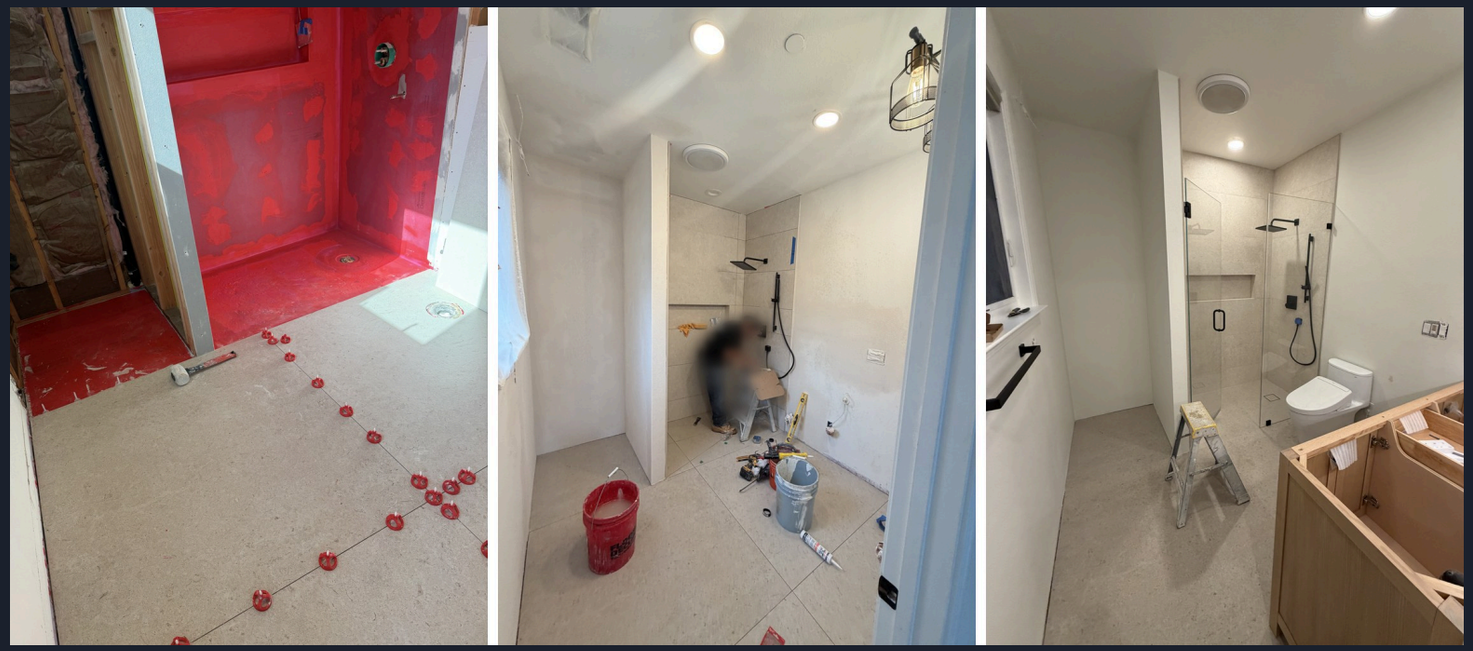
- Where agents pay off outside work, from six things I did this year
- Why the hour you spend first matters more than the model you pick
- Seven principles and practices you can copy

SECTION 1 OF 2

Use Cases

- Supervising a bathroom remodel
- Building a bathroom cabinet
- Installing wall slats in my office
- An app for my Peloton bike
- Reviewing my taxes and financial position
- Tracking my global to-dos

Supervising a bathroom remodel



Gemini helped with the design: my wife used it (Nanobanana and Pro) to render the space and iterate until she had what she wanted. Then sourcing: from a photo of the tile, the stores that carried that exact pattern, down to the aisle.

Normally that means shopping across multiple stores. I also had Gemini produce the installer guide, in English and in Spanish.

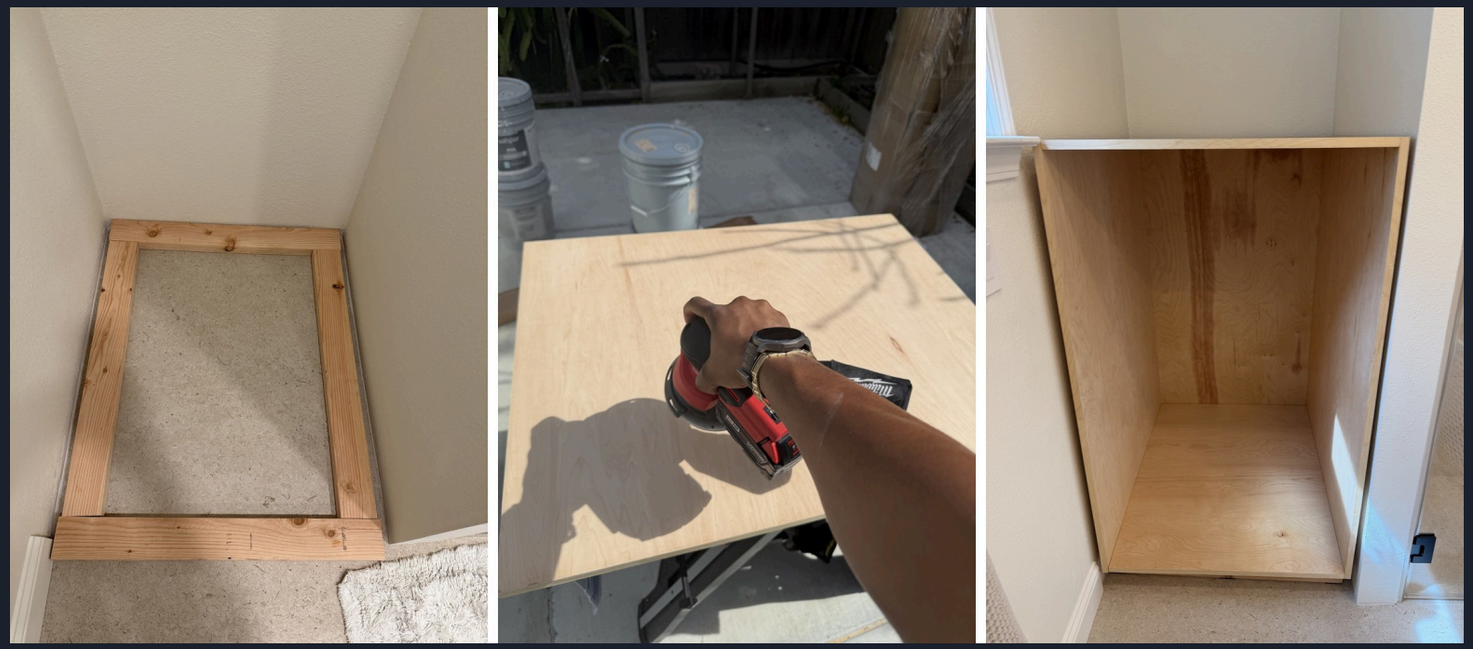
It caught a mistake before it was sealed in the wall

At each stage I sent progress photos and asked what to check before the next one.

Gemini flagged that insulation was missing from a section of wall. Caught before the walls closed, it was fixed early. Later it is far more costly, or never detected.

I was not trying to interfere with the work, only to understand it well enough to ask the right questions.

Building a bathroom cabinet



I had never built one.

Claude helped with navigating build constraints (the cabinet had to be assembled *inside* the bathroom, since the doorway is narrower than the shelf), the joinery, making it moisture resistant, and which tools to buy with no equipment to start.

Claude also makes mistakes

The width of the drawer slides was never carefully calculated. That only showed when it was time to fit the drawers in, and I had to resize them.

It is an iterative process. Nothing is perfect the first time.

Installing wall slats

Claude helped with the product and supplier, comparing several, then the measurements, the cuts, and the installation approach.

It also suggested gluing the panels to the wall. Standard, but it did not match my preference for ease of removal, and I had never said so. I asked what the side effects were and it came back with screws.

Fitness tracking

In 2022 I knew roughly how to do this. It was many weekends of work, so I never started.

- A first version in a weekend: reads the bike's sensor data at 10Hz, tracks sessions, talks to my watch
- Then daily use, with fixes dictated from my phone during the workout
- More consistent in the five weeks since than in the rest of the year

---

Four years of wanting this.

Ridelog
Home
Ride
Workouts
Sessions
Trends
Advanced
Watch
8:15 AM
All
Favorites
Mine
Endurance
Intervals
HIIT
Climb
Test
Recovery
Import
+ New workout
JUST RIDE
1:00:00
Just Ride
No structure — ride as you like. Everything is still recorded.
1 steps
intensity
TEST
1:04:00
FTP Test (20 min)
Classic 20-minute test. FTP = 95% of your 20-min average. Pace it: steady, slightly building,...
12 steps
intensity
ENDURANCE
53:00
Sweet Spot 3×8
Three 8-minute blocks at 88–94% FTP with 4-minute recoveries. The best bang-for-buck...
8 steps
intensity
INTERVALS
52:00
VO2max 5×3
Five 3-minute efforts at 106–120% FTP, 3 minutes easy between. Breathing hard by...
HIIT
24:00
Tabata 2×8
Two sets of 8 × (20 s all-out / 10 s rest), 4 min between sets. Short, brutal, done in 20.
ENDURANCE
45:00
Endurance 45
45 minutes in zone 2 with three short cadence drills. Conversational pace; this is where the...
Sensor
Watch: pairable
Peloton Bike (Gen 1)
10 Hz · 400989 samples
192.168.1.116:8080
v0.2

Reviewing my taxes and financial position

Three years ago, working on AutoGen, we talked about a North Star: the tasks everybody needs to do and nobody wants to do.

I still use TurboTax and a CPA. I checked the agent's numbers against the CPA's. They were the same.

Your CPA can help because they can see all your documents. An agent can help for the same reason. Everything you would hand to TurboTax, in one folder.

Tracking my global to-dos

One markdown file per thread. Frontmatter names the next concrete action, a size, and what it is blocked on. A build script regenerates one board.

Size describes the next action, not the project. A multi-month thread whose next step is a thirty-minute decision is small.

Committed so GitHub renders it, which makes the whole board readable from a phone. Thirty minutes at a swim class is enough to move something.

SECTION 2 OF 2

Principles and Practices

- Start in interview mode
- Assemble the context
- Move between local and remote
- Dictate rather than type
- Supervise the trajectory
- Record recurring notes
- Build strong verification loops

1. Start in interview mode

An hour or two describing the problem and answering questions about it, before anything is built.

- Explore every edge case first
- It becomes a design document in the repo
- I read all of it before anything gets built

In some ways this is similar to emerging practices around spec driven development.

2. Assemble the context

Private repos or folders, organized so an agent can parse and use them.

Git is optional. The organization is not.

The test: could someone else sit down and work from this folder? Setup compounds. Once an expenses repo existed, everything else went under it.

3. Move between local and remote

- **Cloud agents** keep running after you close the laptop. The clean answer, if your org can set it up.
- **Remote control** keeps execution on your machine and moves only the control surface to your phone. That is what I use.

From a phone I can take pictures, connect to my desktop and share updates. That is exactly what made the remodel supervision work.

4. Dictate rather than type

I speak much faster than I type. Most people likely do.

- Makes an hour of interview mode practical at all
- The Claude plugin for VS Code has a clean dictation button; macOS has local dictation too
- You can dictate work from the treadmill or the bike

5. Supervise the trajectory

As you outsource and accelerate the work, ensure you keep the understanding.

- Ask what the side effects are, before you accept the approach
- Read the trajectory that produced the summary

A task whose output you build yourself forces you to understand it. A task whose output is a document lets you skip it.

6. Record recurring notes

- A preference becomes an entry in CLAUDE.md or memory
- A procedure graduates to a skill

My writing process is one. This talk's article was written under it.

7. Build strong verification loops

You cannot read everything, so automate the half a machine can judge.

- Deterministic hooks that run in seconds
- Clean rules rather than opinions: structures you can identify in code, constructions that are just not accepted
- They become the evaluation and verification gate

Two of them checked every paragraph of the article this talk came from. Verification can also be instructions or reusable skills.

The same list, two years earlier

In November 2024 I wrote these practices for software engineering, before any of these projects.

- Analyze first, do not ask for code
- Provide context
- Ask many questions
- Watch for subtle mistakes
- Aim for knowledge parity

---

newsletter.victordibia.com/p/developers-stop-asking-llms-genai

One caution

Agents should work for you. It is easy to invert that.

Letting them take up your attention instead, giving tasks at the risk of your family and the rest of your life.

I wrote about that slope in “/upgrade … or …”, for coding and work. It applies at least as much once the tasks are personal.

Conclusion

Agents can be applied to everyday life tasks, and the quality of life improvement is significant.

The principles that keep you effective here translate back to work.

Starting in interview mode, assembling the context, moving between local and remote, dictating instead of typing, supervising the trajectory, recording what recurs, and building verification loops are the same things that get the most out of agents for work tasks like software engineering.

If you have found ways to reclaim your own time with this, I would like to hear it.