Lecture XII - Project Work II

Programming with Python

Dr. Tobias Vlćek



EP 12: Ship It

One week to launch

Demo Day is **next week**: every group presents in Session XIII. The order comes out of a hat. The hat is Python.

Before you build: ten minutes on making your agent stop repeating the same mistakes.

Teach Your Agent

The friction you know

- Two weeks in, you have corrected your agent on the same things again and again: naming style, the library you prefer, “keep functions short”, “round money to two decimals”
- Every session starts from zero: the agent remembers **nothing** between sessions (Session X, the loop)
- Explaining the same thing five times is work you should not be doing

Package the rules once

- Put your project rules in **one markdown file** in the repo: `AGENTS.md`
- What goes in: what the project is, the tools it uses, naming and style conventions, “any AI use lands in the README disclosure”
- Mistral Vibe reads it at every session start, automatically: write the rules once, stop repeating them

Skills: instructions on demand

- A **skill** is the next step up: a folder with a `SKILL.md`, holding a name, a description, and instructions
- Rules apply **always**. A skill loads only when the task matches its description, so context stays lean until the moment it is needed
- Example: a `verify` skill for your project: read the diff, run it, test it on a case where you know the answer, report what you checked
- In your stack: `.vibe/skills/verify/SKILL.md` in the repo. Vibe loads it in the terminal and inside Zed’s agent panel, same engine

- Vibe follows an open format, so skills you write here also work in other agents that adopt it

Anatomy, briefly

```
---
name: verify
description: Check changed code the way this project requires. Use after any
code change.
---

- Read the diff line by line before running anything
- Run the code on a case where you know the correct answer
- Report what you checked and what the result was
```

Plain markdown you own, at `.vibe/skills/verify/SKILL.md`, versioned with the project like any other file.

Write your own

- Any workflow you have explained to the agent **twice** deserves a skill
- Keep it short: a page of instructions, not an essay
- Prune it when the workflow changes: a stale skill misleads
- Downloading one? Vet it by **reading it**, not by how popular it is

What not to hand over

- The parts you must be able to **defend at the demo**: design choices, the statistics, why this approach
- A file can carry conventions. It cannot carry your judgment, and the demo questions come to you, not to it

Definition of Done

Before you call it finished

- The repo runs from a fresh clone (cloned once more into an empty folder): `uv sync`, then your script
- The README says what it does, how to run it, and carries the **AI disclosure section**
- The commit history tells the story: small steps, messages that mean something
- `git status` is clean: all committed, nothing important ignored
- I am invited, and the link is on Moodle before your demo

> **Tip**
> Run the fresh-clone test **today**, not on demo morning. It always finds something.

Demo Day Is Next

Next week's format

- **7 minutes** presenting, **3 minutes** of questions, per group
- Order: drawn at random with the `random` module from Session VI
- Show the thing running, the commit history, and one thing that broke
- Everyone in the group has to be there

The rest of today is the last supervised work session: I am in the room for blockers and "is this scope still realistic?"

Wrap-up

Three things to remember

- Rules in `AGENTS.md` for the **always** stuff, skills for the on-demand stuff, both plain markdown you own
- The fresh-clone test is the cheapest bug you will ever catch
- A demo is the thing running, the history, and the disclosure

Next week: **Demo Day**. The investor is in the audience.

Bibliography