Code with AI

How-To and Good Practices

Dr. Hugo BACARD - CNRS/IJCLab

S3 School 2026, Annecy

S3 School

Who am I?

- Current role: AI engineer at CNRS/IJCLab (EVERSE)
- Background: Mathematician (Algebra, Homotopy)
- Goal: To **share with you** how I use AI for coding.

Preliminary Warnings

- ✓ Different LLMs ⇒ Different Outputs

- ✓ Better LLMs ⇒ Better Outputs

- ✓ **Your prompts matter:**
  Good prompts ⇒ Good outputs

- ✓ **No Unique Solution:**
  Unless you structure the outputs + Constraints on LLM settings

Preliminary Warnings

⚠️ **Always Be Critical:**

- AI models have hallucinations
- AI can invent non-existant libraries or packages
- Always Test
- No Blind Trust

Preliminary Warnings

Security and Ownership Considerations

⚠️ Ensure you have permission to use external AI for your project.

⚠️ Copyrights issues with code generated by AI. (You should check before)

Requirements: AI Resources

Requirements: AI Resources

Need an LLM with good coding capabilities

Requirements: AI Resources

Need an LLM with good coding capabilities

💻 LOCAL

LLM runs on your machine

☁️ CLOUD

Sovereign Resources

Not Sovereign (Paid or Free)

Chat Interface or
ApiKey + endpoint + model

Running LLMs Locally

?

Ollama

Ollama is a tool that allows to you run, manage, LLMs on your local machine.

Ollama

Ollama is a tool that allows to you run, manage, LLMs on your local machine.

⚡Easy Setup

Simple CLI commands to download and run models

💻 Local First

Works offline, Complete privacy

🔧 Based on llama.cpp

Optimized inference engine in C++

🌐 https://ollama.com/

https://github.com/ollama/ollama

Ollama: Installation (Optional)

Linux

curl -fsSL https://ollama.com/install.sh | sh

macOS and Windows

Available Apps on GitHub: https://github.com/ollama/ollama

Main CLI commands

- ollama pull model_name
- ollama run model_name

Details on GitHub https://github.com/ollama/ollama

Cloud Resources

Browser-based Chat Interface

- ChatGPT
- Claude
- Le Chat
- Gemini
- DeepSeek

API Based Usage

Requires:

- ApiKey
- Endpoint
- Model

Advice

Advice

Use AI to find the right tools for your work.

Advice

Use AI to find the right tools for your work.

Ask Questions (and Verify)

- How do I run LLMs on a Mac (Linux, Windows) ?
- What are the open-source alternatives to [something]?
- I am Phd Student with a Linux machine, how do I run LLMs locally using open-source technology? Provide a step-by-step guide.

AI Resource: Codestral from Mistral AI

AI Resource: Codestral from Mistral AI

Create an account at https://codestral.mistral.ai/

(Phone number verification required)

Generate an ApiKey (and save it!)

- Endpoint: https://codestral.mistral.ai/v1
- Model: codestral-latest

Other « Free » Resources:

https://github.com/cheahjs/free-llm-api-resources?tab=readme-ov-file

AI Coding Assistance: Main Approaches

AI Coding Assistance: Main Approaches

IDE + Browser-based AI

- ChatGPT 🇺🇸
- Claude 🇺🇸
- Le Chat 🇫🇷
- Gemini 🇺🇸
- DeepSeek 🇨🇳

IDE + Plugin or CLI app

- Copilot (⚠️ Now supports Ollama models)
- Open-source alternatives: Continue, Cursor, Windsurf…

IDE + Browser-based AI

- ChatGPT 🇺🇸
- Claude 🇺🇸
- Le Chat 🇫🇷
- Gemini 🇺🇸
- DeepSeek 🇨🇳

✅ Pros:

- Access to better models
- Good for prototyping
- Models with long contexts

IDE + Browser-based AI

- ChatGPT 🇺🇸
- Claude 🇺🇸
- Le Chat 🇫🇷
- Gemini 🇺🇸
- DeepSeek 🇨🇳

✅ Pros:

- Access to better models
- Good for prototyping
- Models with long contexts

❌ Cons:

- Privacy & Security Concerns
- Costs (for paid plans)
- Your prompts may be used to train models
- Manual copy-pasting
- **Requires Internet!**

IDE + Plugin (extension)

VS Code Extensions	Local/Sovereign LLMs support
GitHub Copilot (⚠️ Microsoft)	✅ (⚠️)
Cursor	✅
Continue	✅
Windsurf	✅

IDE + Plugin (extension)

VS Code Extensions	Local/Sovereign LLMs support
GitHub Copilot (⚠️ Microsoft)	✅ (⚠️)
Cursor	✅
Continue	✅
Windsurf	✅

✅ Pros:

- Privacy with Local or Sovereign AI
- Integrated within your workspace
- No costs with local models
- Context-aware (knows your whole project)
- Works offline with local models

IDE + Plugin (extension)

VS Code Extensions	Local/Sovereign LLMs support
GitHub Copilot (⚠️ Microsoft)	✅ (⚠️)
Cursor	✅
Continue	✅
Windsurf	✅

✅ Pros:

- Privacy with Local or Sovereign AI
- Integrated within your workspace
- No costs with local models
- Context-aware (knows your whole project)
- Works offline with local models

❌ Cons:

- Local models don’t match with paid
- **Copilot can scan your docs, apiKeys, etc**
- Costs with paid AI models
- More hallucinations with small models

Install Continue

VS Code and JetBrains Plugin

- Integrates AI assistance directly in your IDE.
- Open-source alternative to GitHub Copilot
- Support for custom LLMs.

- **Website**: https://www.continue.dev/
- **GitHub**: https://github.com/continuedev/continue
- **Install**: Search for Continue in the market
- **Docs**: https://docs.continue.dev/

Add Codestral in Continue

Locate the YAML config file:

- **Linux/macOS**: ~/.continue/config.yaml
- **Windows**: %USERPROFILE%\.continue\config.yaml

Add Codestral in Continue

Locate the YAML config file:

- **Linux/macOS**: ~/.continue/config.yaml
- **Windows**: %USERPROFILE%\.continue\config.yaml

Add in the models block:

```
models
    -name: Codestral
    provider: mistral
    model: codestral-latest
    apiKey: your-secret-api-key
    apiBase: https://codestral.mistral.ai/v1
    roles:
      - autocomplete
      - chat
      - edit
```

Documentation for different AI providers:

https://docs.continue.dev/customize/model-providers/overview

Key Aspects of AI-Assisted Coding

Code Understanding	Code Debugging	Code Translation
Code Improvement	Code Completion	Vibe Coding

Code Understanding

Use AI to understand a complex or « messy » code

Exercise 1

Open a web browser and choose an AI chat (ChatGPT, Claude, Le Chat) and try the prompt templates below with the main.py of the pkoffee project.

> **PROMPT 1**
> Explain the code below to an engineer discovering it.
> CODE:

> **PROMPT 2**
> Explain what this code is doing conceptually, how the main components interact, and what problem it is solving. Assume the reader is technically competent but unfamiliar with the codebase. CODE:

> **PROMPT 3**
> Explain this code at a high level for an engineer who didn’t write it. Focus on intent, data flow, and design choices rather than line-by-line mechanics.CODE:

Add Prompts in Continue

Add a « prompts » field in the config.yaml and save your templates so you can reuse them (Don’t Repeat Yourself).

```
prompts:
    - name: unit-test
      prompt: >
        Provide a comprehensive set of unit tests using pytest that cover all the functionality of the code below. Explain why the tests are relevant and what they are testing.
- name: CodeRefactor
    prompt: >
      Refactor the code below to improve its quality, performance, and maintainability. Provide explanations for the changes made.
```

💡Add the previous prompts on Code Understanding in Continue ;-)

Code Understanding: Tips 💡

Effective Prompting Strategies

Purpose and context	• What problem does this code/script solve ? • What background is required to understand the purpose of this code?

Code Understanding: Tips 💡

Effective Prompting Strategies

Purpose and context	• What problem does this code/script solve ? • What background is required to understand the purpose of this code?
Your role / background	I am a [role] discovering this code, my task is to [your_task]. Identify the most important parts to examine first.

Code Understanding: Tips 💡

Effective Prompting Strategies

Purpose and context	• What problem does this code/script solve ? • What background is required to understand the purpose of this code?
Your role / background	I am a [role] discovering this code, my task is to [your_task]. Identify the most important parts to examine first.
High Level Structure	• What are the main stages or components in this [script/code]? • What is the data flow from input to output? • Ask AI for documentation generation

Code Understanding: Tips 💡

Effective Prompting Strategies

💡 My prompts are usually vague, I use AI to refine them.

💡 Submit the code to the LLM and ask for the best prompt for your task !

⚠️ Before submitting, make sure no private data or proprietary code is present!

Code Debugging

Provide the AI with code containing errors and prompt it to identify and suggest potential solutions for the identified issues.

Example of Prompt for Debugging

Example

The below [language] [language_version] code gives this error:<content_of_error>

Debug what's wrong and explain how I can improve the code.

<content_of_code>

Example of Prompt for Debugging

Example

The below [language] [language_version] code gives this error:<content_of_error>

Debug what's wrong and explain how I can improve the code.

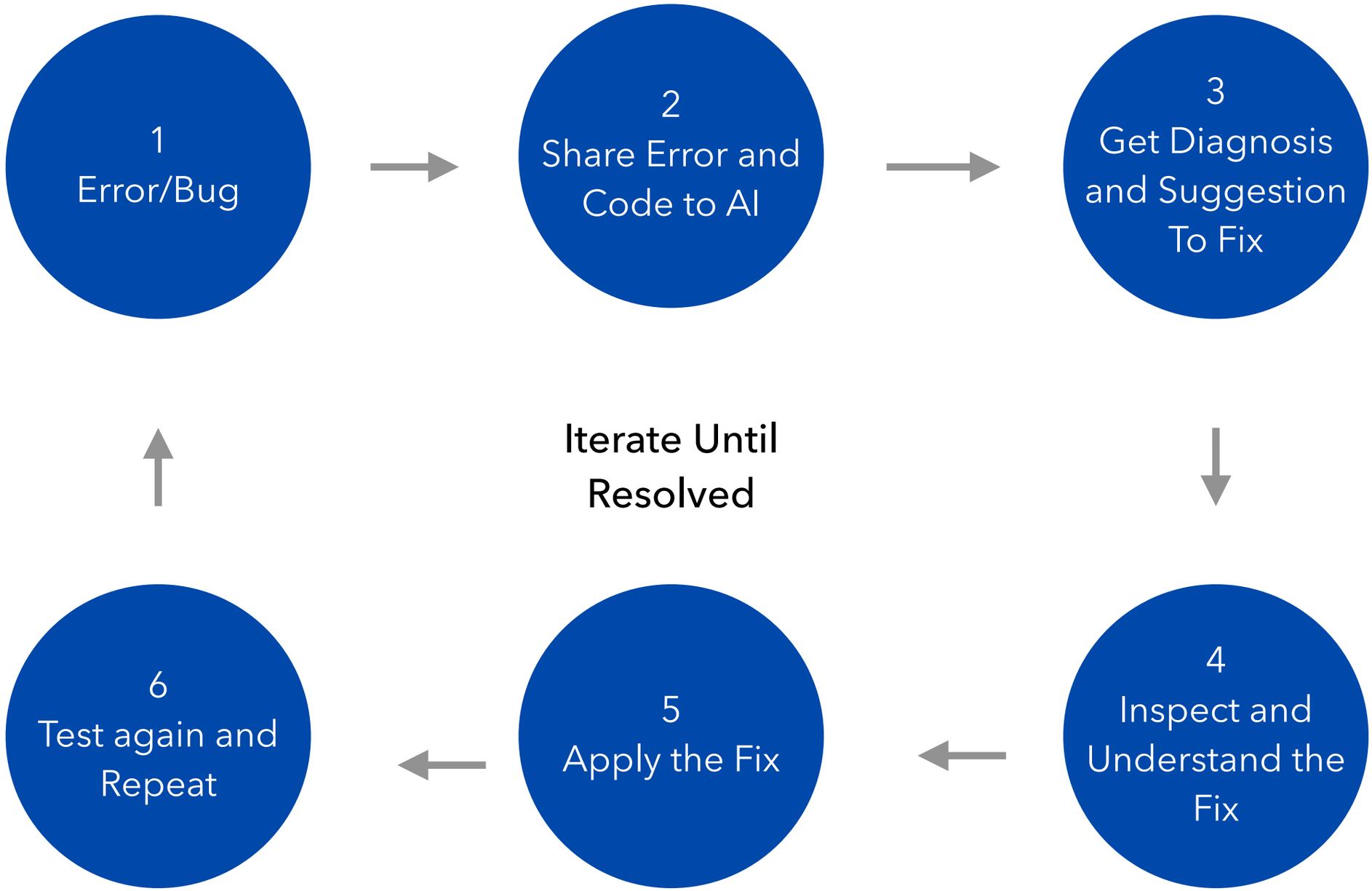
<content_of_code>

Sources:

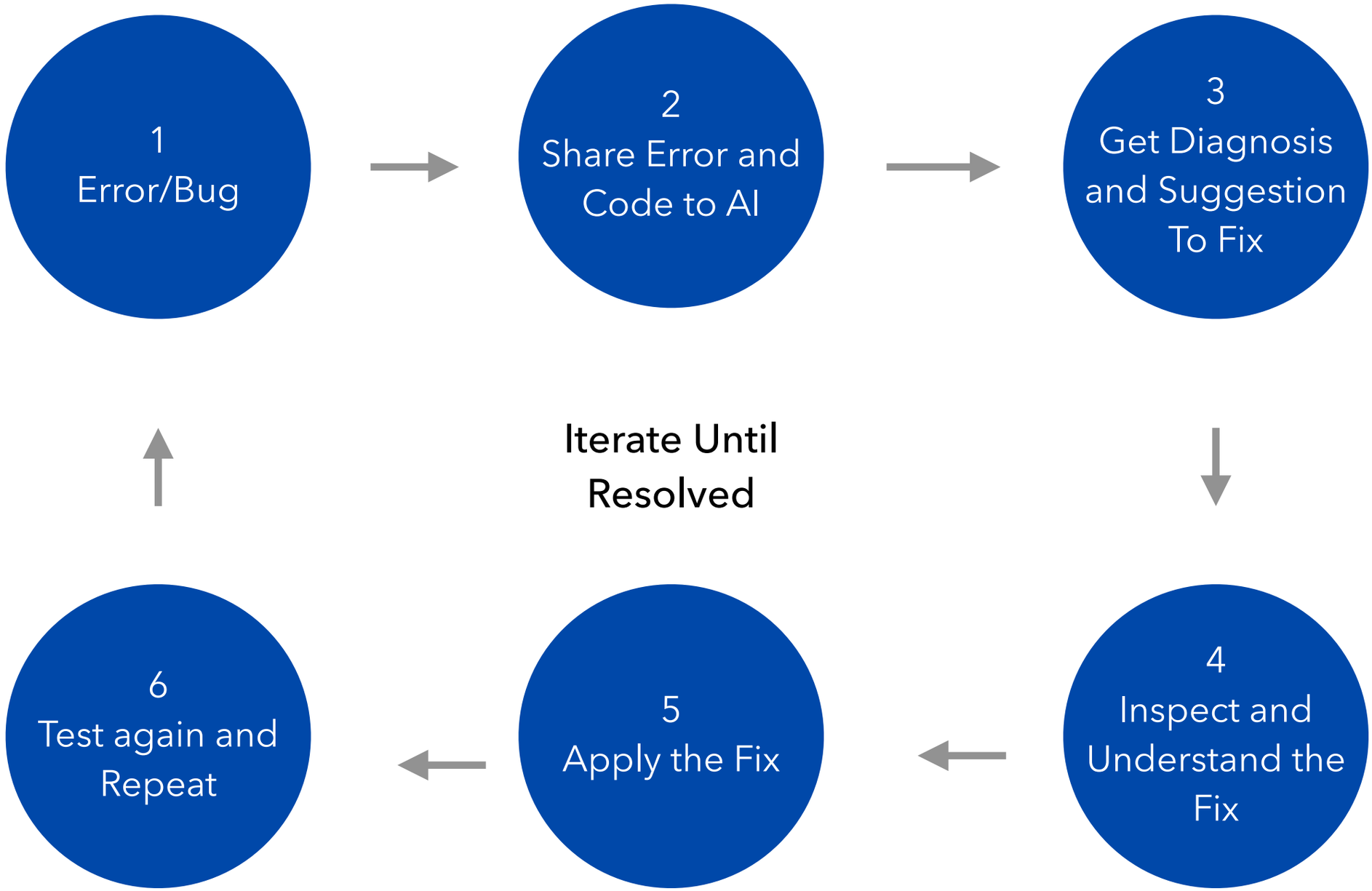
- https://cloud.google.com/discover/what-is-prompt-engineering
- https://www.kaggle.com/whitepaper-prompt-engineering

⚠️ Watch out for context length

Iterative Debugging Loop



Iterative Debugging Loop



⚠️ LLMs can be easily « distracted » when multiple contexts are mixed (it will miss the real bug)

Exercise 2

> Use AI to debug the codes located here:
>
> **https://github.com/s3-school/s3-2026-lectures/blob/main/docs/4.2-ai-assisted-development/exercise_2.md**

Code Debugging: Tips 💡

DO NOT ❌	DO ✅
Use prompt like « Fix this code »	Describe the exact bug or error
Dump all your code	Share the minimal failing snippet
Assume context	State language, version, framework
Change everything at once	Apply one fix at a time
Trust blindly	Ask why the fix works

Code Debugging: More Tips 💡

Add Persona in your prompts	Act as a Senior Python Developer …
Add Constraints	Fix the bug without changing the API signature …
Output	Explain the fix before showing the code

Code Improvement + Refactoring

Use AI to improve, refactor or optimize your base code.

Exercise 3

Try the prompt templates below with the main.py of the pkoffee project.

PROMPT 1	CODE: <content_of_the_code> Improve the above code to make it robust, maintainable and easy to read. Explain each modification and why.

PROMPT 2	Act as an experienced Google engineer and review the following Python code snippet. Provide suggestions for improvements to enhance maintainability, concentrating on readability, modularity, and best practices. Offer concise explanations for each suggested change to facilitate easy integration: CODE:

Question: Which one worked ``better'' ? Which AI provider do you prefer ?

Code Translation

Specify the source and target programming languages to enable the AI to translate code while preserving functionality and syntax.

Exercise 4

> Use AI assistance to translate into Python the codes located here:
>
> **https://github.com/s3-school/s3-2026-lectures/blob/main/docs/4.2-ai-assisted-development/exercise_4.md**

```rust
fn main() {
    let f = |x: f64| x.powi(2) - x;
    let a = 0.0;
    let b = 1.0;
    let n = 1000;
    let w = (b - a) / n as f64;
    let mut t = 0.0;
    for i in 0..n {
        let x = a + i as f64 * w;
        t += f(x) * w;
    }
    println!("The result is: {}", t);
}
```

Code Translation: Advices

- Be the judge: Only translate into a language you really understand!
- 💡Code understanding with AI –> Then translate with your understanding
- Make sure you have permission to translate the code
- Always Test!
- ⚠️ **AI can generate malicious code that can damage your system/ database** 😈

Code Completion

Code completion is a feature in many IDEs that provides suggestions for completing code as you type.

Code Completion

Benefits ... ?

- **Faster development**: Reduces typing and speeds up coding by suggesting functions, variables, and syntax.
- **Fewer errors**: Helps prevent typos and syntax mistakes by offering valid, context-aware options.

Code Completion

Benefits ... ?

- **Faster development**: Reduces typing and speeds up coding by suggesting functions, variables, and syntax.
- **Fewer errors**: Helps prevent typos and syntax mistakes by offering valid, context-aware options.

Available with ... ?

GitHub Copilot, Continue, Cursor, Windsurf, …

Exercise 5: Code Completion

1. Start typing a python function that returns the list of all prime numbers less than a given integer n.

2. Ask AI to act as Google engineer to improve the previous code.

3. Evaluate and test the code. Is it still working ? Is the code really improved? How do you know it’s better?

4. If not already present, ask for error handling, proper typing.

5. Do you agree ? Inspect carefully.

6. Ask for unit tests following Max’s advices. Ask why are the suggested tests relevant.

Good practice: Divide and Conquer

Good practice: Divide and Conquer

1

Clearly Identify the Scope

- Do you need a package? A standalone script ? An application ? Fast Prototyping
- In Which Language ? Which Framework ?

Good practice: Divide and Conquer

1

Clearly Identify the Scope

- Do you need a package? A standalone script ? An application ? Fast Prototyping
- In Which Language ? Which Framework ?

2

Use AI to get the Structure / Architecture

Example: ```Describe a modern repository structure that would be suitable for the script below as its main function. Consider factors like code organization, testing, documentation, and CI/CD pipelines. CODE:```

Good practice: Divide and Conquer

1

Clearly Identify the Scope

- Do you need a package? A standalone script ? An application ? Fast Prototyping
- In Which Language ? Which Framework ?

2

Use AI to get the Structure / Architecture

Example: ```Describe a modern repository structure that would be suitable for the script below as its main function. Consider factors like code organization, testing, documentation, and CI/CD pipelines. CODE:```

3

Implement, Test, Debug, Integrate

- Start with independent components/modules
- Tests and integrate

Exercise 6: Fast Prototyping

1. Use the main.py from the pkoffee project to generate a repo structure.
2. Once you get the structure, ask the AI to provide a Python script that creates it.
3. Use the generated repo structure along with the main.py script to ask the AI how to structure the code
4. Ask the AI for guidance on adopting the right mindset for generating tests for this project. (Act as a Google engineer ...)
5. Ask the AI to assess potential security issues for this project.
6. What are the available tools for security audit for this project? (Might invent tools)

Advice

Advice

Use AI to Refine/Reformulate Your Prompts

Example: (You want unit tests)

- « What prompt would a Google engineer use to write unit tests for this code using pytest? »
- Evaluate the prompt and Start a new fresh conversation with the refined version.

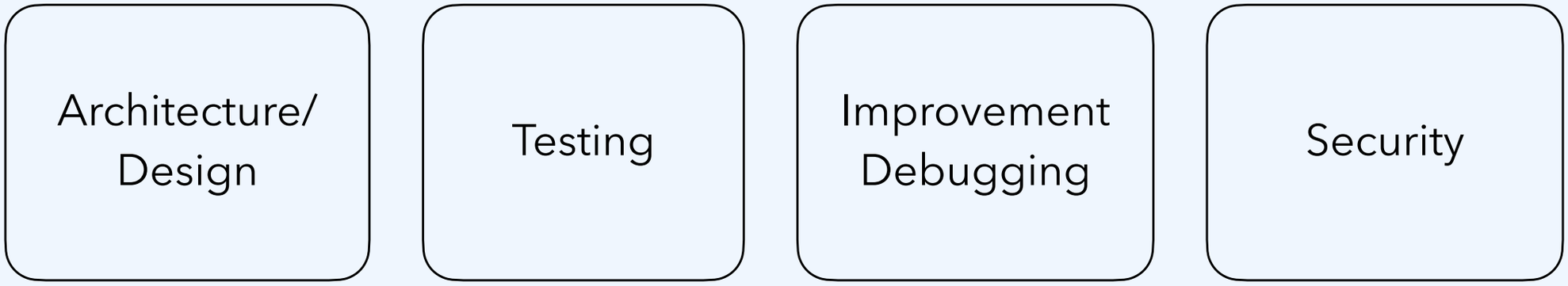
Advice

Use AI to Refine/Reformulate Your Prompts

Example: (You want unit tests)

- « What prompt would a Google engineer use to write unit tests for this code using pytest? »
- Evaluate the prompt and Start a new fresh conversation with the refined version.

Build Your Prompt-Space



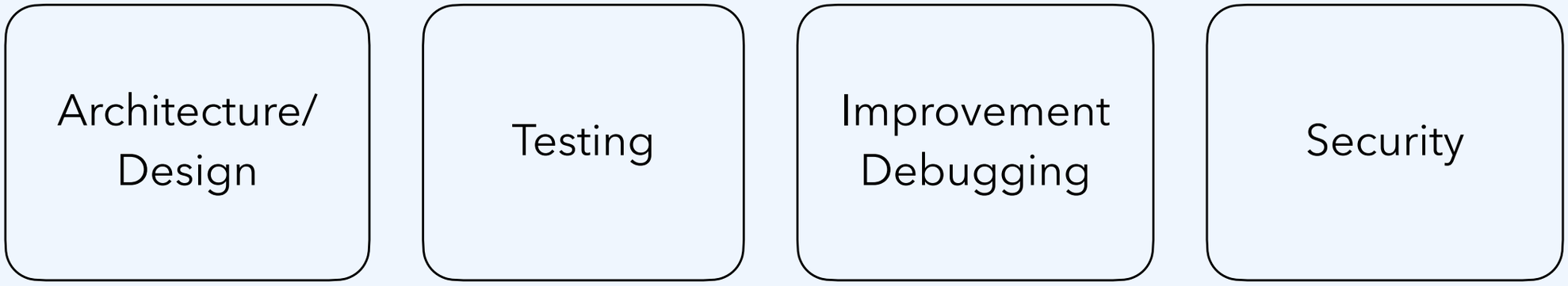
Advice

Use AI to Refine/Reformulate Your Prompts

Example: (You want unit tests)

- « What prompt would a Google engineer use to write unit tests for this code using pytest? »
- Evaluate the prompt and Start a new fresh conversation with the refined version.

Build Your Prompt-Space



Use AI to find the Right Tools

AI can connect to Internet. Ask AI to identify the best tools on the market for your project.

Add an MCP Server in Continue

Add an MCP Server in Continue

Objective

Add Context7 in Continue

(Same procedure with other MCP servers)

Add an MCP Server in Continue

Objective

Add Context7 in Continue
(Same procedure with other MCP servers)

What is ... ?

- Context7
- An MCP server ?
- MCP ?

MCP

In Short

- MCP = Model Context Protocol.
- An open-source standard for connecting AI applications to external systems.
- Developed by Anthropic (https://modelcontextprotocol.io/)
- Think of it as a USB port for AI applications

MCP

Benefits

Using MCP, AI can connect to:

- Data sources (files, databases)
- Tools (Search engine, calculator, data analysis script,… pkoffee MCP?)
- Shared tools and services within a team/institution

Context7

In Short

Context7 is an MCP server that provides the latest documentation to AI models so they generate accurate, up-to-date code and answers.

https://context7.com/

Install in VS Code:
https://context7.com/docs/resources/all-clients#vs-code

Exercise 7: Use Context7

> Use AI assistance to fix the code here:
>
> **https://github.com/s3-school/s3-2026-lectures/blob/main/docs/4.2-ai-assisted-development/exercise_7.md**

Implement a simple MCP server

- Use the Python SDK (FastMCP) to implement an MCP server that returns the list of all prime numbers less than a given number.
- Add it to Continue

Docs: https://github.com/modelcontextprotocol/python-sdk

Security and Data Privacy

- Always ensure your institution allows using [external] AI services.
- Always anonymize data before sending your prompts to a non sovereign AI.
- Never paste API Keys
- Use Local AI for sensitive work

Security and Data Privacy

- ⚠️ Code Debugging: Traceback call errors may expose the home directory structure or sensitive information

- ⚠️ GitHub Copilot: Do not leave the .env in the workspace as it will be scanned!

- ⚠️ AI might suggest deprecated or vulnerable packages!

Security and Data Privacy

- ⚠️ Review all AI-generated code (Malicious code snippets)
- ⚠️ DO NOT USE CODE YOU DON'T UNDERSTAND!
- ⚠️ Only use trusted MCP servers (High vulnerability)

Exercise 8: Security (Optional)

Try with different AI the prompts below with the main.py script

> PROMPT 1
> Identify security issues in this code

> PROMPT 2
> Are there any unsafe operations or potential vulnerabilities?

Question: Which one is better ? Do you understand the output?

Exercice 9

1. Implement a simple retriever using BM25s (pip install bm25s) https://github.com/xhluca/bm25s
2. Use pymupdf to implement a pdf reader.
3. Implement a text chunker that creates chunks using chunk_size and chunk_overlap
4. Create a pipeline that reads a pdf file, creates chunks and loads into the retriever.
5. Implement a Streamlit-based UI for retrieving and displaying the top_k most relevant chunks to a given query.
6. Add an LLM from Mistral to create a RAG chatbot experience.

Key Takeaways

Be the Thinker.
AI is a tool.
Use AI to reformulate.
Good Prompt = Good Output

Use AI to find the best coding tools for your work

Always Review and Test AI-generated Code

Security and Privacy Matter

Use AI to Learn

Iterate. Don’t expect perfection

Open Discussion: Vibe Coding ?

« A sort of definition »: Coding by describing the « vibe » or intent and letting AI handle the syntax entirely.

The Question	Are we becoming "System Architects" rather than "Coders"?
The Risks	If the AI makes a mistake in the "vibe," can you still fix it manually?
Is it the future?	Let’s discuss!

Exercise 10: Vibe code pkoffee!

1. Use the main.py to find a good prompt for rebuilding the pkoffee project from scratch.

2. Use AI to vibe code it. Evaluate.

Thank You.