CEC Continuing Education Center

DesignNews

Creating Your Own AI Software Intern

DAY 3 : Customizing AI Agents into Your Workflow Part 1

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

CEC Continuing Education Center

BENINGO EMBEDDED GROUP

DigiKey

THE SPEAKER

Jacob Beningo

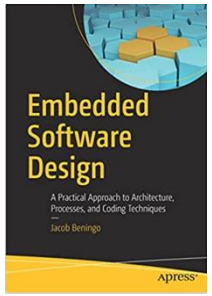
Jacob@beningo.com

jacobbeningo

Beningo Embedded Group – CEO / Founder

Focus: Software Architecture, Processes, and Dev Skills

At Beningo Embedded Group, we believe everyone deserves the skills to confidently advance their careers, meet deadlines, and deliver quality embedded systems. We provide modern strategies, insights, and hands-on training to equip developers and teams with the tools they need to succeed.



EMBEDDED SOFTWARE ACADEMY BY BENINGO

Visit **www.beningo.com** to learn more

Agentizing your GPT

01

Agentizing Your GPT

You already have a GPT that:

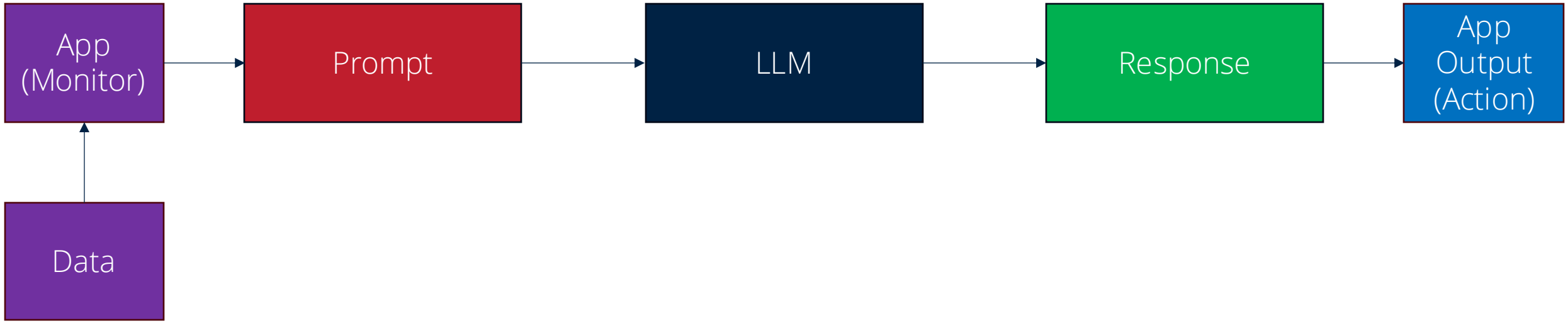
- Analyzes code
- Follows a style guide
- Adds Doxygen comments

Now ask: What should it do on its own?

Example Agent Role:

- "Monitor a directory. When a .cpp file appears or is modified, review and document it automatically using our style guide, and output results to a new file."

Agentizing Your GPT

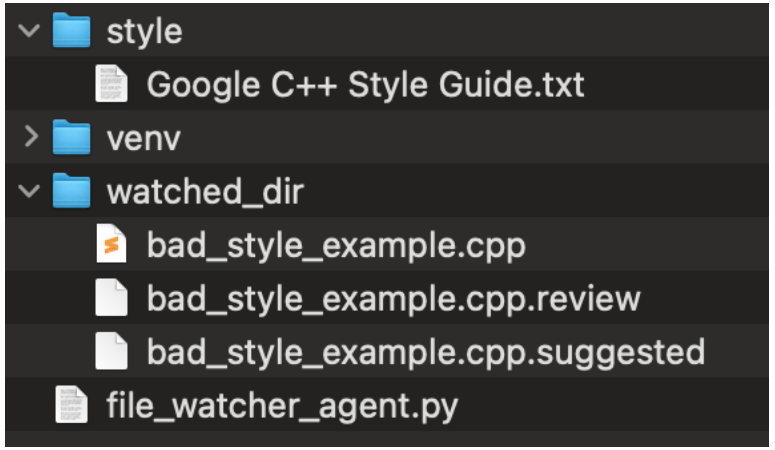


Agentizing Your GPT

Build an Agent using Python that does the following

- Monitors a directory for change
- On a change generates a prompt to analyze the file(s) using:
  - Our coding style
  - The files code
- Uses the analysis response to:
  - Create a review file
  - Create a suggested changes file

Example Test Directory Result:



Audience POLL Question

How much time over a year would an agent like this save you?

a) None, it would cost me more time
b) 0 – 80 hours
c) 80 – 160 hours
d) 160+ hours

The Change Monitor

02

The Change Monitor – Environment Setup

```
# Step 1: Create a virtual environment
python3 -m venv venv

# Step 2: Activate it
source venv/bin/activate

# Step 3: Install watchdog inside it
pip install watchdog
```

The Change Monitor – Environment Setup

```python
# file_watcher_agent.py
import time
import os
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

WATCH_DIR = "./watched_dir"  # Change this to your folder

class CppFileHandler(FileSystemEventHandler):
    def on_modified(self, event):
        if event.src_path.endswith(".cpp"):
            print(f"[MODIFIED] {event.src_path}")
            process_file(event.src_path)

    def on_created(self, event):
        if event.src_path.endswith(".cpp"):
            print(f"[CREATED] {event.src_path}")
            process_file(event.src_path)

def process_file(filepath):
    # Placeholder: replace this with GPT API call
    with open(filepath, 'r') as f:
        code = f.read()

    print(f"\n🔍 Reviewing {os.path.basename(filepath)}...\n")
    print(f"--- Begin File ---\n{code}\n--- End File ---\n")
    print("✍️ Simulated GPT Review: 'Add Doxygen comments and fix formatting...'\n")

    # Optional: write output to .review file
    with open(filepath + ".review", 'w') as f:
        f.write("Simulated GPT Review:\nAdd Doxygen comments and fix formatting...")
```

```python
if __name__ == "__main__":
    event_handler = CppFileHandler()
    observer = Observer()
    observer.schedule(event_handler, WATCH_DIR, recursive=False)
    observer.start()
    print(f"👀 Watching for changes in {WATCH_DIR}...")

    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        observer.stop()
    observer.join()
```

The Change Monitor – Test

Test Steps

- Start the Monitor Agent
- Add a new .cpp file
- Modify the .cpp file

Verify that the watchdog detects new and modified files. →

```
👀 Watching for changes in ./watched_dir...
[MODIFIED] /Users/jacobbeningo/00-agent/watched_dir/bad_style_example.cpp

🔍 Reviewing bad_style_example.cpp...

--- Begin File ---
#include<iostream>
using namespace std;
int addnumbers(int a,int b){
return a+b;
}

int subtractNumbers(int a,int b)
{return a-b;}

int Multiply_Numbers( int a , int b ){
    return a*b;}

int divideNumbers(int a,int b){if(b==0)return 0;return a/b;}

float average(int a,int b)
{
return (a+b)/2.0;
}



--- End File ---

📣 Simulated GPT Review: 'Add Doxygen comments and fix formatting...'
```

Audience POLL Question

Do you have an OpenAI API account?

a) Yes
b) No
c) Working on it . . .
d) other

Adding the LLM

03

Adding the LLM – OpenAI SDK Setup

Install the OpenAI SDK.

- pip install openai

Get your OpenAI API key:

- Go to https://platform.openai.com/account/api-keys
- Copy your key

Adding the LLM – Adding to the Agent

- In a terminal, add your key to your environment:

```
export OPENAI_API_KEY='sk-...'
```

- Access the key in your python script:

```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")
```

Adding the LLM – Test the Connection

```python
import time
import os
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler
import openai

# Set OpenAI API client (openai >= 1.0.0)
client = openai.OpenAI(api_key=os.getenv("OPENAI_API_KEY"))

WATCH_DIR = "./watched_dir"
STYLE_GUIDE_PATH = "./style/Google C++ Style Guide.txt"
```

```python
def process_file(filepath):
    try:
        with open(filepath, 'r') as f:
            code = f.read()

        print(f"🔍 Reviewing and suggesting improvements for {os.path.basename(filepath)}...")

        style_guide = load_style_guide()

        # Step 1: Review
        review_output = call_gpt(REVIEW_PROMPT, code, style_guide)
        with open(filepath + ".review", 'w') as f:
            f.write(review_output)
```

Audience POLL Question

Does adding an LLM to an application seem easier or more difficult than expected?

a) As expected
b) More difficult
c) other

Next Steps

04

Going Further

Download the Agent Sourcecode:

- https://beningo.short.gy/l6ykgr

- Attend my EOC workshop
- Review ChatGPT Documentation
- Agent Documentation
- AI Cookbooks

Additional Resources

Please consider the resources below:

- Jacob's Blogs
- Jacob's CEC courses
- Embedded Software Academy

- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

Consulting | Coaching | Training

www.beningo.com

Next Steps

- Introduction to AI-Powered Software Development
- Customizing Your AI Intern with GPTs
- Integrating AI Agents into Your Workflow Part 1
- Integrating AI Agents into Your Workflow Part 2
- Deploying and Optimizing Your AI Intern

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets