



COT6930-001 – Gen AI in Software Development Lifecycles

Instructor: Dr. Fernando Koch

Assignment 1 – Hands-On Prompting Engineering for Requirement Analysis

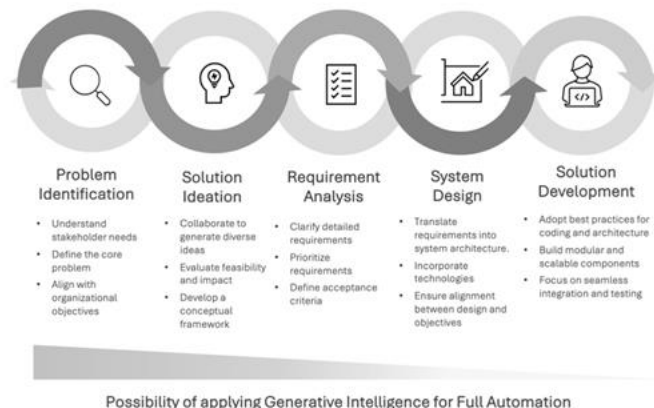
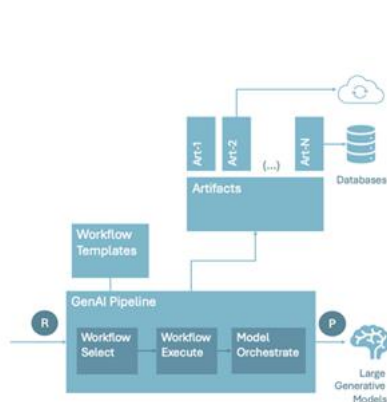
Objective:

Expose students to hands-on Prompt Engineering through a simple programming exercise. Students will setup their Lab Environments, getting exposure to Ollama, Prompt Engineering, and coding of GenAI Pipelines. The goal is to promote hands-on experiences with diverse techniques around Prompt Engineering and get them thinking different about Prompts, Pipelines, and Models.

By the end of this exercise, everyone should have:

- Ollama installed on his computer or learned how to use remote Ollama
- Experience with basic Model Request and what does it mean ‘talk to the model’
- Hands-on experience with Prompt Engineering Techniques and coding for Prompt Engineering
- Experience with coding Automated methods for SDLC

Arguments:



From the GenAI point of view:

- GenAI is about Prompts, Pipelines, and Models.



- This include process of Prompt Engineering (creating the right prompt), Automated Prompt Engineering, Workflow Selection, Workflow Execution, and Model Orchestration.
- In some configurations, it may also include calling ‘artifacts’ to execute external code, like retrieving entries from databases, collecting information from other systems or the Web, and others.
- Diverse techniques of Prompt Engineering, including Automated Prompt Generation (where the user asks the GenAI model to generate a prompt for a specific problem – increases the quality of the results significantly).
- There is no complete science about how to compose the best prompt for the job.
- Instead, there is a combination of experience, critical thinking, proper automation, and experimentation.
- There are numerous **combinations of Prompt Engineering techniques to Automated Prompt Engineering for a given Task**, including combinations of:
 - Prompt Engineering Techniques
 - Levels of Automation
 - Variations of Model Parameters
 - Chaining Automation and Manual Prompting adjustment
 - Others

From the SDLC point of view:

- From the SDLC point of view, GenAI delivers better cost-benefit results in the early stages of Software Development Lifecycles, in helping solution developers to conceive and structure new solutions.
- During these initial stages, the thought process is less structure and the support to idea and exploration is more evident, should the user apply the right techniques to obtain the desired results from GenAI models.
- As the tasks become more structured towards Solution Design and Developer, the quality, impact, and efficiency of the results become less evident.
- Requirement Analysis sits in the middle, where solution developers need to bridge the gap between free thinking and action.
- We argue that well-crafted Prompt Engineering can lead to superior results during the requirement Analysis phases.
- For the automation of this process, there is a demand to balance between synthetic creativity, solution alignment, control, and efficiency.

Leading research question:

- Which configuration of Prompt Engineering works the best **combination of techniques to Automated Prompt Engineering** for a the task of **Requirement Analysis in SDLC?**



Your Task

- Install the provided **Prompt Engineering Lab** on your computer / combination of computers.
- Devise an experimental Use Case:
 - What is the Solution you want to build?
 - How to automated the process of Requirement Analysis for the implementation of this solution?
- Execute experiments with coding prompting techniques, applying the **Prompt Engineering Lab**, test variations of Prompts, Techniques, and Model Parameters that attain the best results (i.e. the best answers for the Requirement Analysis task)
- The **README.md provides the section 'Experimenting'** with instructions on what can be modified in the code

<https://github.com/genilab-fau/prompt-eng>

click on README.md

Expected Outcomes

1. **Experiments with Prompt Engineering Techniques**, including (pick what you like):
 - a. Experimenting with different Prompt Templates for Requirement Analysis to attain best accuracy.
 - b. Completing the code for the Prompt Engineering Technique that you want to experiment
 - c. Creating a new code for other Prompt Engineering Technique
 - d. Experiment with combinations of Automated Prompt Generation in a chained process
 - e. Experiment with combinations of Prompt Engineering Technique, Automated Prompt Generation, and variation of Model Parameters, to identify best performance both for latency and accuracy.
 - f. (other ideas welcomed; the sky is the limit)

More ideas for research and explorations in the List of Research Ideas at:

<https://github.com/genilab-fau/prompt-eng>

click on CONTRIBUTING.md

2. A short **Research Report about** your experiments and the results you have achieved



- template provided in **MY-REPORT.md in your installation**
- the latest release will be at:

<https://github.com/genilab-fau/prompt-eng>

click on MY-REPORT.md

3. Resulting code for the Use Case and Prompt Engineering that you have devised through the **Prompt Engineering Lab**

4. **SUBMISSION:**

- Complete and test your code
- Fill out **MY-REPORT.md in your installation**
- Create YOUR Github REPO with your CODE
- Submit the **URL to YOUR Github Repo** through CANVAS

Note:

- Good Code and Good Use Cases will be invited to integrate the new related of the Prompt Engineering Lab
- Good Research Reports will be invited to extend and internal publication
- Exceptional Research Reports will be invited to extend and external publication in a venue to be defined; to be exceptional:
 - Experiments to generate data to support comparative analysis between method, techniques, and variations of parameters.
 - Good visualization e.g. diagrams and graphs presenting the experiments and results.

General Instructions

- **For COT6930-2xx Professional Program: INDIVIDUAL Assignment**
- **For COT6930-001 and -002: GROUP Assignment**
- **To complete the Assignment**, Submit the **URL to your Github Repo** through CANVAS
- **Use the Prompt Engineering Lab and Research Report template being provided.**
- **You are free to Define the Use Cases for your Requirement Analysis.**
- **Define an interesting use case** and follow the SDLC technique: Problem Identification, Solution Ideation, then Requirement Analysis.



- Keep in mind the target is to **AUTOMATE the process of Requirement Analysis**; not the requirement analysis per se!
- **Use GenAI/ChatGPT/other** extensively during this process; that is what this course is about.
- **For simple ideas** on how to complete the assignment and help out :

Check the list of **!!!!IMMEDIATE HELP NEEDED!!!** in CONTRIBUTING.md

How to get it done?

- (1) Install the **Prompt Engineering Lab** on your computer as described at:

<https://github.com/genilab-fau/prompt-eng>

- Check the **Troubleshooting page** if you are facing installation or execution problems:

<https://github.com/genilab-fau/prompt-eng>

click on TROUBLESHOOTING.md

- Validation: your environment will be working once you can execute one of the ‘ready to use’ techniques being provided under /prompt-eng folder (zero_shot, few_shots, etc)

- (1) **Prepare your code** to experiment with Prompting Engineering Techniques

- Check the example code being provided under /prompt-eng folder (zero_shot, few_shots, etc)
- There are only a few places you can alter the code; the documentation is provided within the code; this simplification should **make your exercises very specific and easy to execute**.
- The **possibilities to prepare your code** include:
 - a. **If you decide to just alter existing techniques**, change the Prompt Templates and execute the process to verify the results.
 - b. **If you decide to complement a provided technique**, alter the code as needed, adjust the Prompt Templates, and execute the process.
 - c. **If you decide to create a new technique (!)**, make a copy of an existing technique (copy & paste the file) that is better aligned with technique you want to create, alter the code as needed, adjust the Prompt Templates, and execute the process.



- d. **If you want to go further:** same process, make a copy of an existing technique as a template, alter, adjust, and execute.

Check for ideas in **Experiments with Prompt Engineering Techniques** (see Expected outcomes) and from the list of Research Ideas in:

<https://github.com/genilab-fau/prompt-eng>

click on .md

Check the list of **!!!!IMMEDIATE HELP NEEDED!!!** in CONTRIBUTING.md