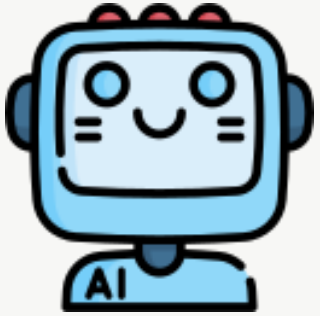


香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

CSC4801
AI-assisted Software Engineering

Lecture 06: Requirements

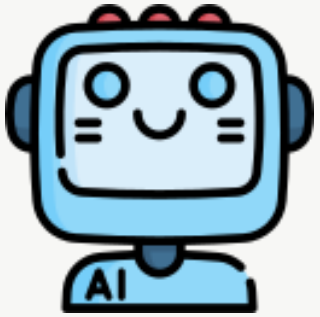
Instructor: Jinsheng Ba | 2026 Fall



Agent



Toy project



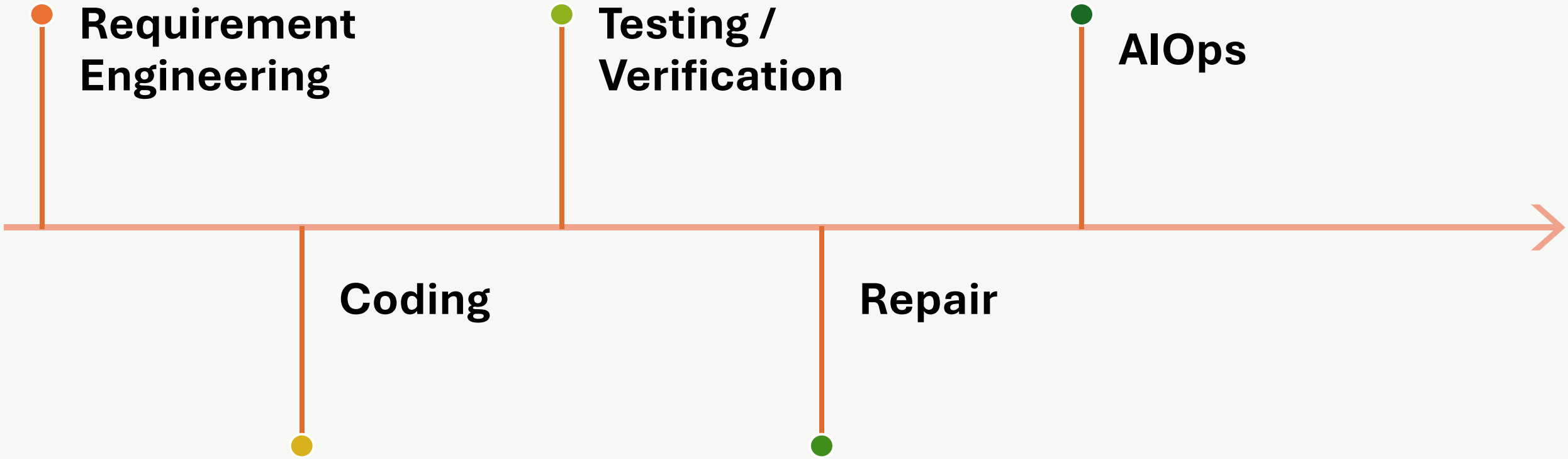
Agent

Software Engineering

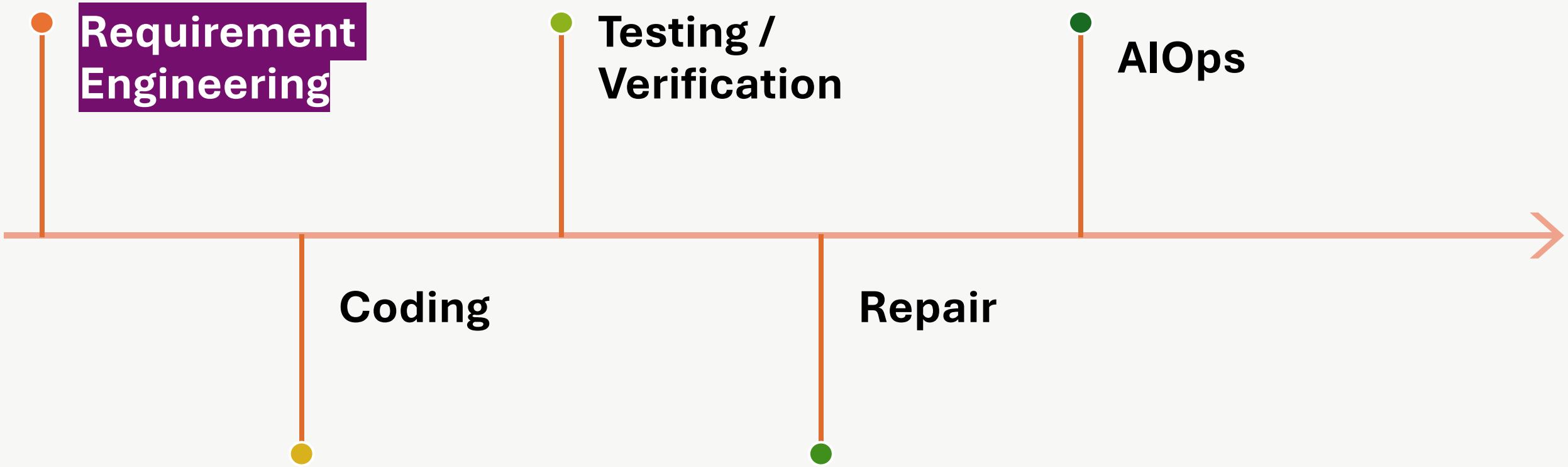


Real-world complex projects

AI-Assisted Software Engineering



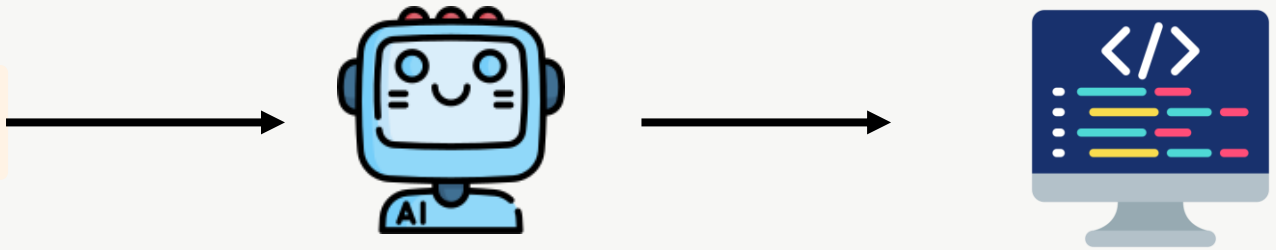
AI-Assisted Software Engineering



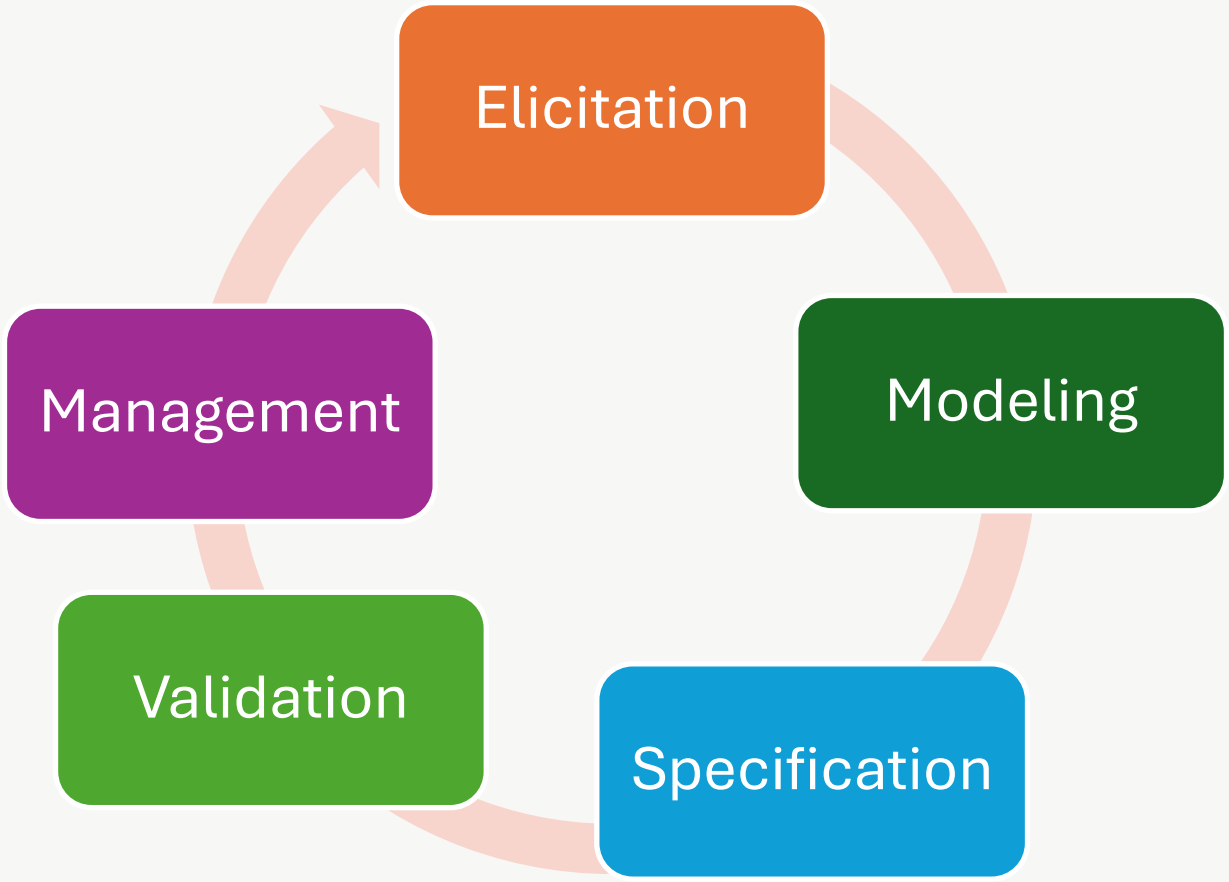
Vibe Coding

- Vibe coding is cool
- How to handle complex systems?

Implement a 2048 game.



Requirement Engineering



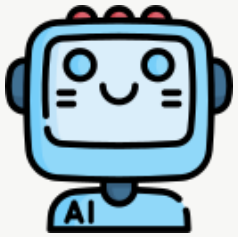
Elicitation

AI extracts latent needs from unstructured meeting transcripts and user review.

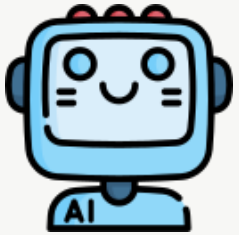
Elicitation

We could use agents to simulate users to understand requirements.

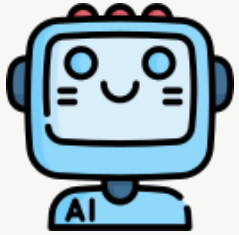
Multi-agent Collaboration Strategy



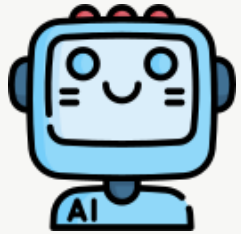
Agent 1



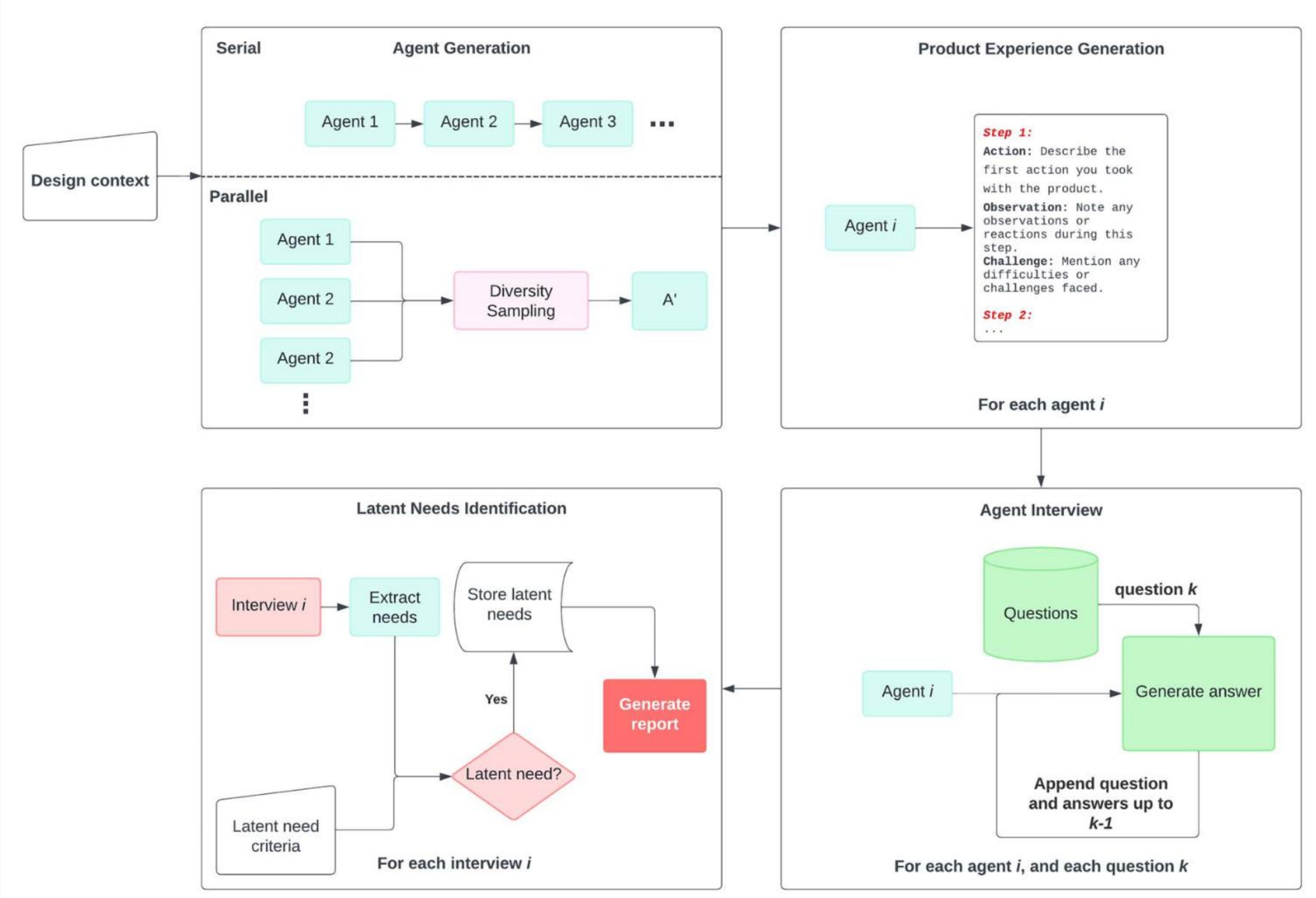
Agent 2



Agent 4



Agent 3



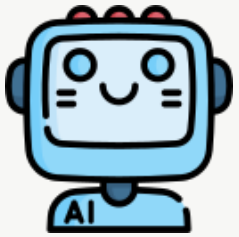
https://arxiv.org/pdf/2404.16045

Multi-agent Collaboration Strategy



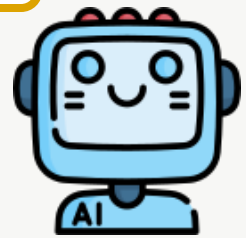
Elevator control system

Interview



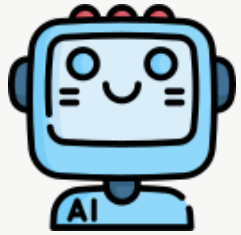
Children

Interview



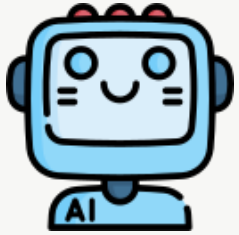
Managers

Interview



The aged

Interview



Workers

Elevator System

REQ1. When the user presses the "Up" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving upwards if necessary, and open its doors upon arrival.

REQ2. When the user presses the "Down" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving downwards if necessary, and open its doors upon arrival.

REQ3. When the user presses any floor button inside the elevator, the Elevator System shall prioritize servicing the selected floor, moving upwards or downwards as needed, and open its doors upon arrival.

REQ4. When the overload sensor detects excessive weight in the elevator cabin, the Elevator System shall prevent further entry, emit an audible alarm, and display an overload warning. It shall not move until the excess weight is reduced and remain in the "Overload" state until the weight is within the acceptable limit.

REQ5. When the user presses any floor button inside the elevator while the system is in the "Overload" state, the Elevator System shall ignore the button press until the overload condition is resolved.

Modeling

AI identifies entities/relationships to construct UML or formal specifications.

How to Translate?

Elevator System

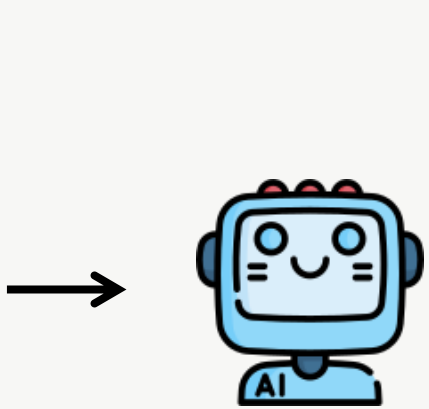
REQ1. When the user presses the "Up" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving upwards if necessary, and open its doors upon arrival.

REQ2. When the user presses the "Down" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving downwards if necessary, and open its doors upon arrival.

REQ3. When the user presses any floor button inside the elevator, the Elevator System shall prioritize servicing the selected floor, moving upwards or downwards as needed, and open its doors upon arrival.

REQ4. When the overload sensor detects excessive weight in the elevator cabin, the Elevator System shall prevent further entry, emit an audible alarm, and display an overload warning. It shall not move until the excess weight is reduced and remain in the "Overload" state until the weight is within the acceptable limit.

REQ5. When the user presses any floor button inside the elevator while the system is in the "Overload" state, the Elevator System shall ignore the button press until the overload condition is resolved.







Conversation

Agent

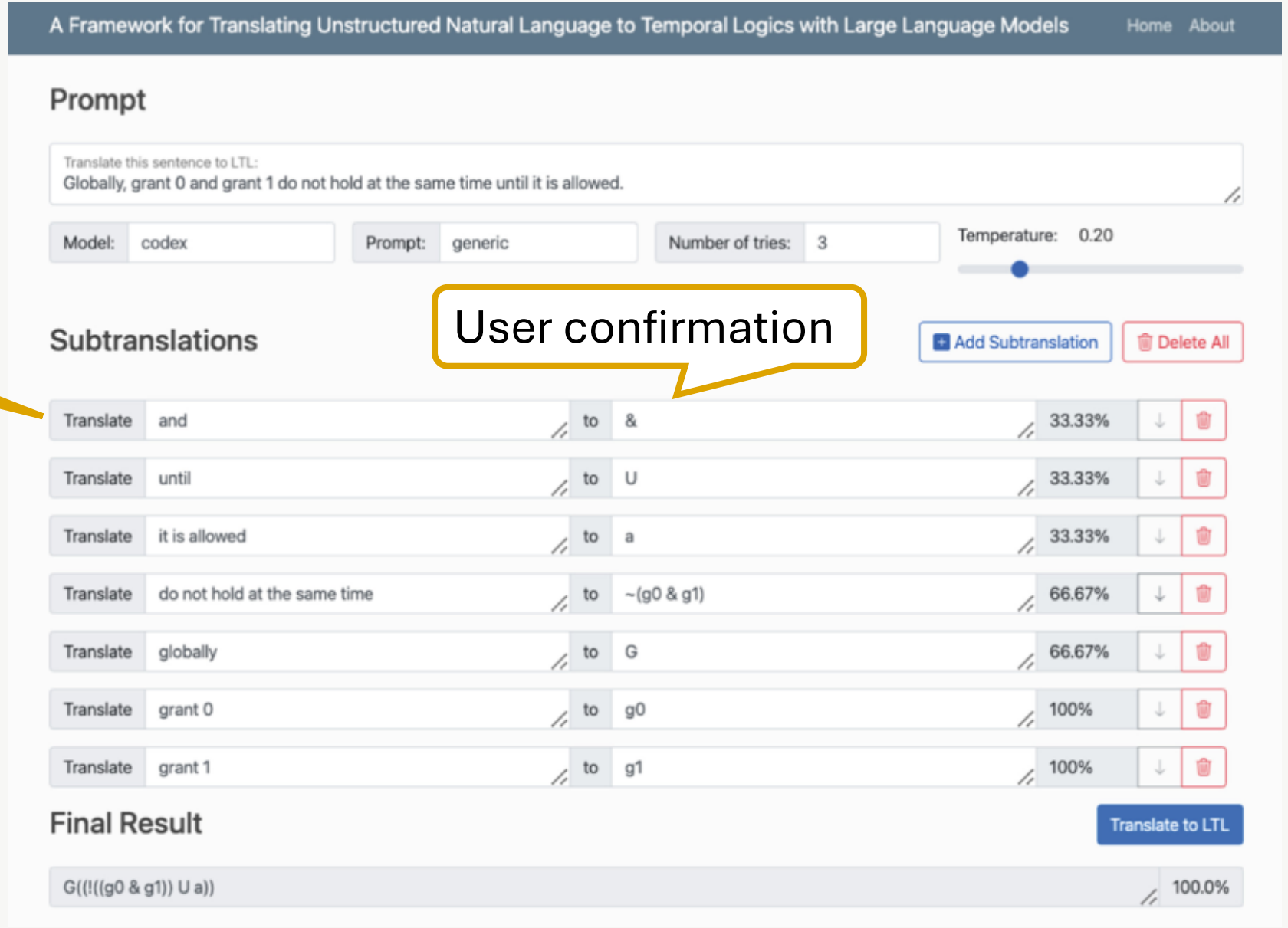
Specifications

Linear Temporal Logic

- LTL (Linear Temporal Logic) is a way to write rules about how a system behaves over time.

- Imagine a traffic light. Ordinary logic can say, “the light is red now.” LTL can say things like:

- G red → … — “Whenever the light is red, …” (G means always)
- F green — “The light will eventually be green” (F means eventually)
- G (red → X green) — “Whenever it’s red, it’s green in the next step” (X means next)



Decompose

User confirmation

https://github.com/realChrisHahn2/nl2spec
https://arxiv.org/abs/2303.04864

Model Checker

- Give the checker two things:

- A model of the system: its possible states and transitions.
- An LTL property, such as G (request → F response) (“every request eventually gets a response”).
- The checker explores the model’s possible executions. If the property fails, it returns a counterexample: an execution showing how it can fail.

Lean4

- **Lean 4** is a programming language and **proof assistant**. You can use it to write mathematical statements or software specifications, then write proofs that Lean checks for correctness.

```
fun max (a, b):
    if a > b then a else b
```

- Specification: the result must be at least as large as both inputs.

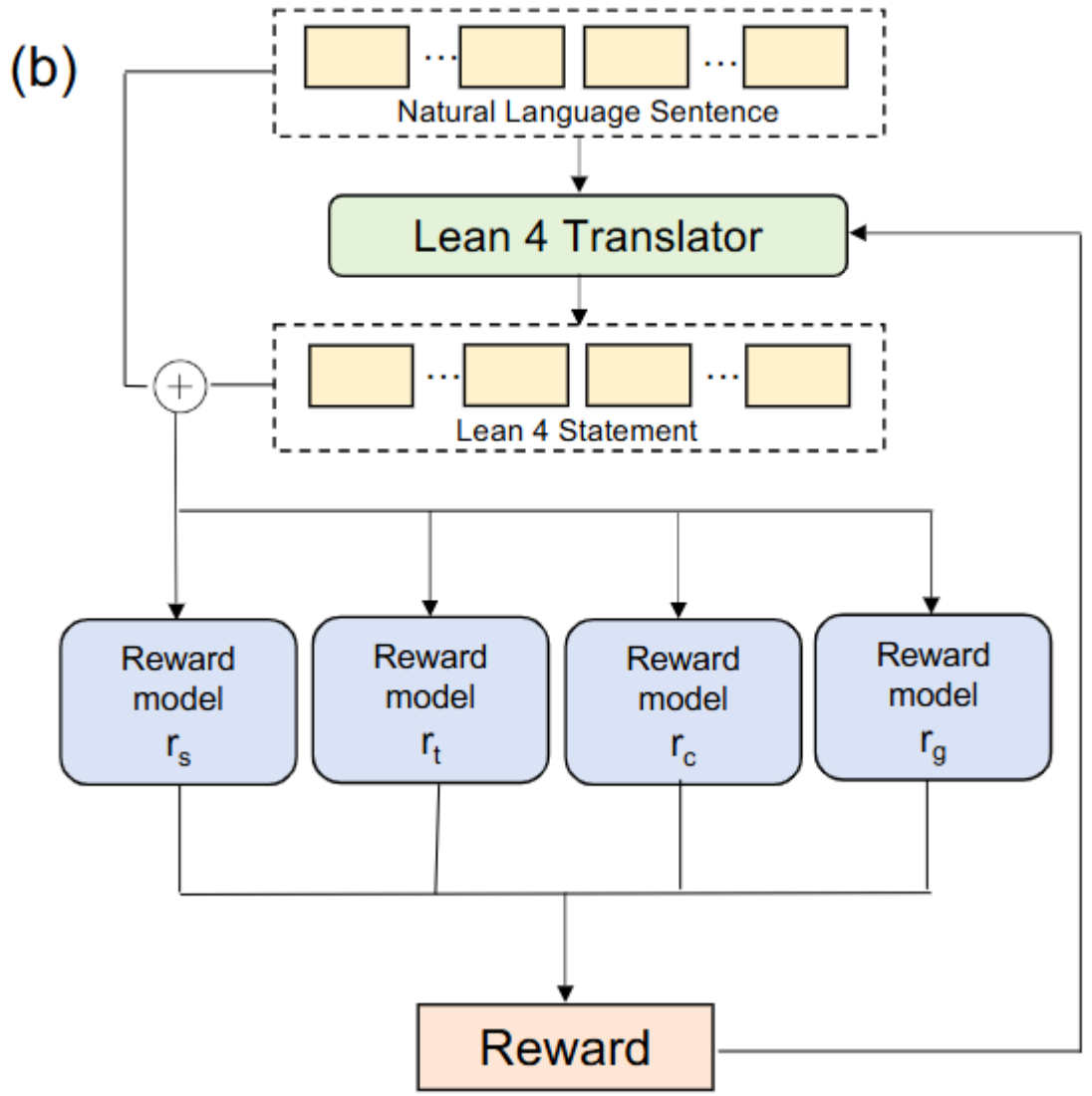
```
def IsMaximum (a b result : Nat) : Prop :=
    a ≤ result ∧ b ≤ result ∧ (result = a ∨
result = b)
```

1. a ≤ result: the result is at least a.
2. b ≤ result: the result is at least b.
3. result = a ∨ result = b: the result is one of the inputs.

NL2Lean: T **4 through**

Yue Fang[1,2*], S **ihan Zhang[3]**
Weiwei **Jin[1,2†]**
[1]School China
[2]Key Laboratory), MOE, China

y.fang@stu.pk @pku.edu.cn



Specification

Requirements are determined and documented.

Elevator System

REQ1. When the user presses the "Up" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving upwards if necessary, and open its doors upon arrival.

REQ2. When the user presses the "Down" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving downwards if necessary, and open its doors upon arrival.

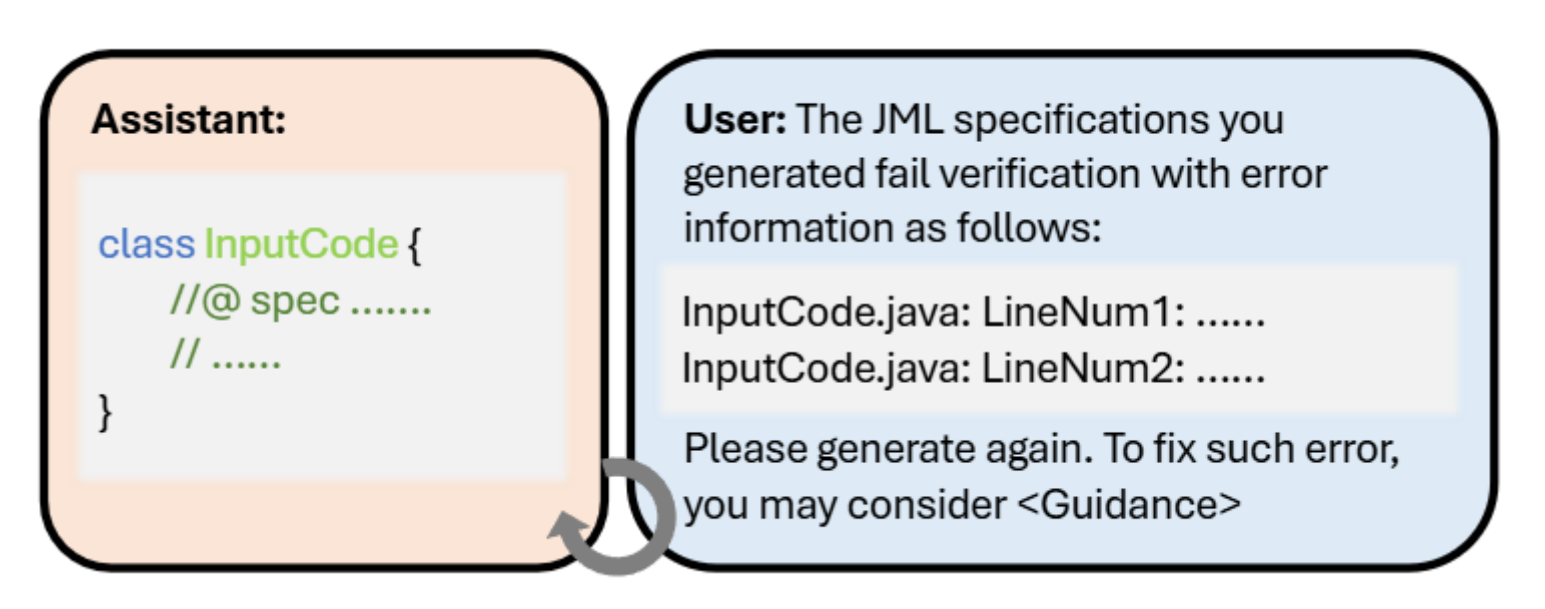
REQ3. When the user presses any floor button inside the elevator, the Elevator System shall prioritize servicing the selected floor, moving upwards or downwards as needed, and open its doors upon arrival.

REQ4. When the overload sensor detects excessive weight in the elevator cabin, the Elevator System shall prevent further entry, emit an audible alarm, and display an overload warning. It shall not move until the excess weight is reduced and remain in the "Overload" state until the weight is within the acceptable limit.

REQ5. When the user presses any floor button inside the elevator while the system is in the "Overload" state, the Elevator System shall ignore the button press until the overload condition is resolved.



Conversation Agent Specifications Implementation



Directly call LLMs

Multiple Tries for Generation

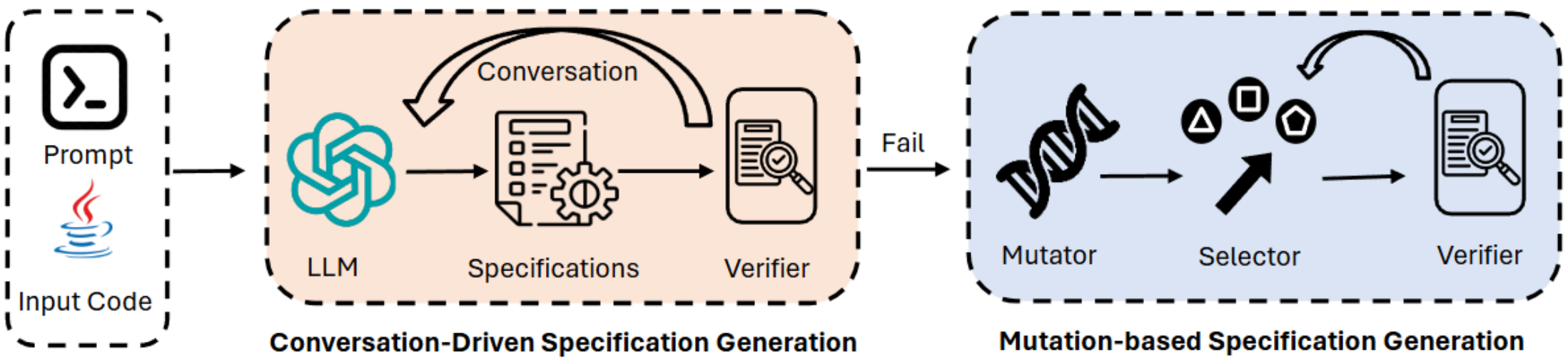


Fig. 2: Overview of our *SpecGen*.

https://arxiv.org/pdf/2401.08807

Verification

AI scores requirements against rules.

Elevator System

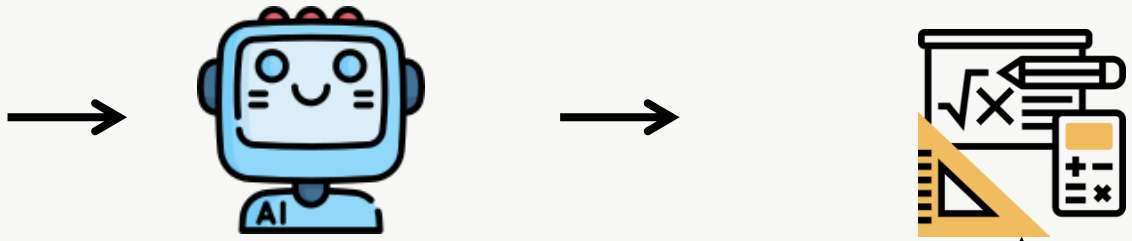
REQ1. When the user presses the "Up" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving upwards if necessary, and open its doors upon arrival.

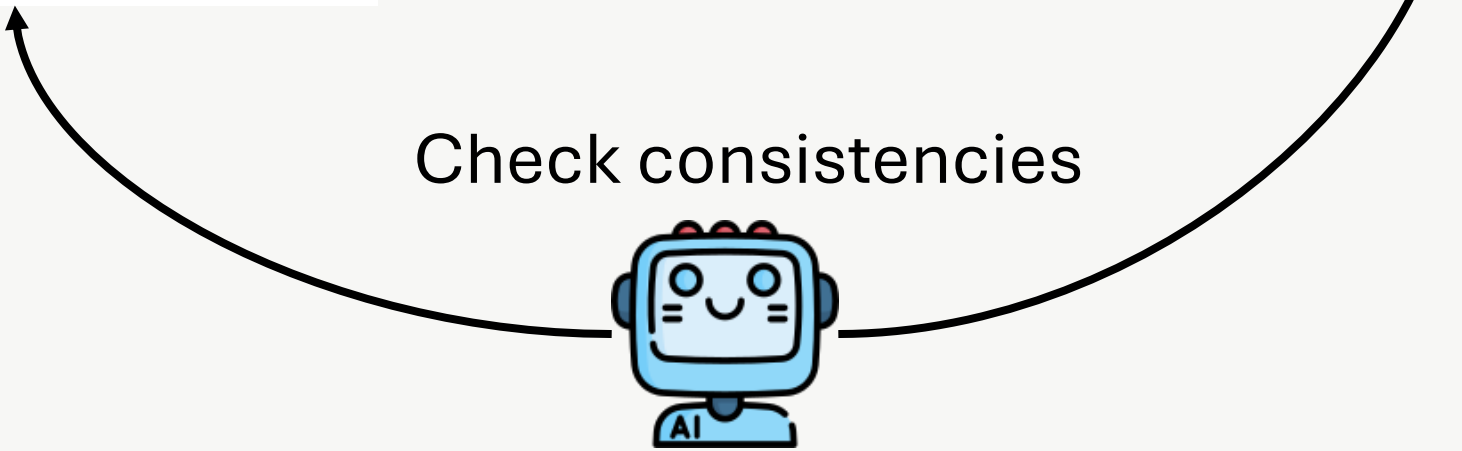
REQ2. When the user presses the "Down" button on a floor, the Elevator System shall prioritize servicing the requested floor, moving downwards if necessary, and open its doors upon arrival.

REQ3. When the user presses any floor button inside the elevator, the Elevator System shall prioritize servicing the selected floor, moving upwards or downwards as needed, and open its doors upon arrival.

REQ4. When the overload sensor detects excessive weight in the elevator cabin, the Elevator System shall prevent further entry, emit an audible alarm, and display an overload warning. It shall not move until the excess weight is reduced and remain in the "Overload" state until the weight is within the acceptable limit.

REQ5. When the user presses any floor button inside the elevator while the system is in the "Overload" state, the Elevator System shall ignore the button press until the overload condition is resolved.





Fantechi, Alessandro, et al. "Inconsistency detection in natural language requirements using chatgpt: a preliminary evaluation." *2023 IEEE 31st International Requirements Engineering Conference (RE)*. IEEE, 2023.

Comparable Performance to Human

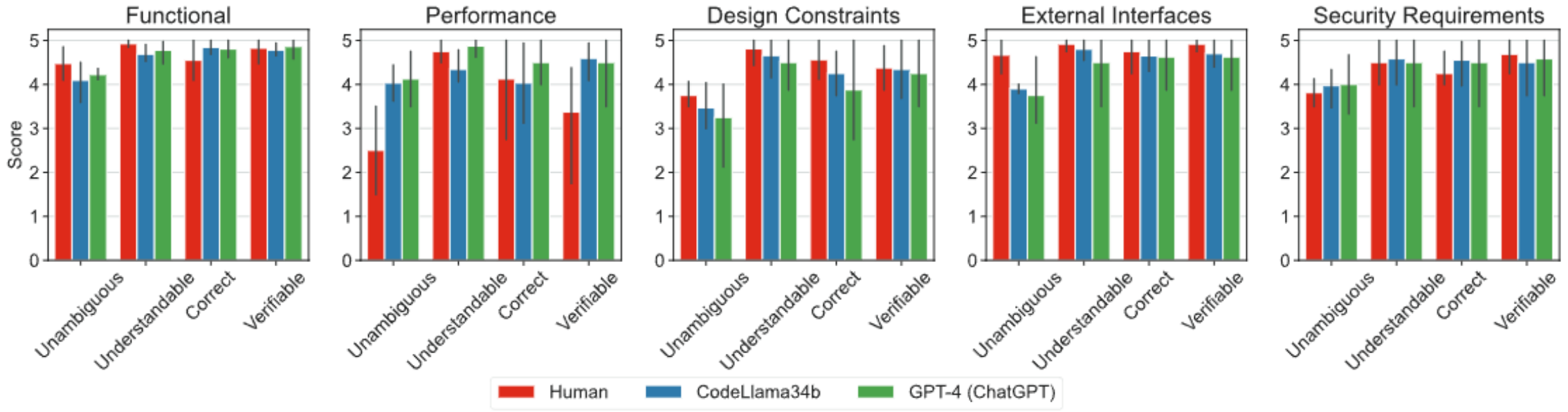


Fig. 3. Per-requirement evaluation results for the three SRS documents. Each graph corresponds to each section of the SRS and has been obtained by averaging the ratings provided by human graders for that part of the SRS.

An evaluation of LLM4RE
https://arxiv.org/pdf/2404.17842

Full Pipeline

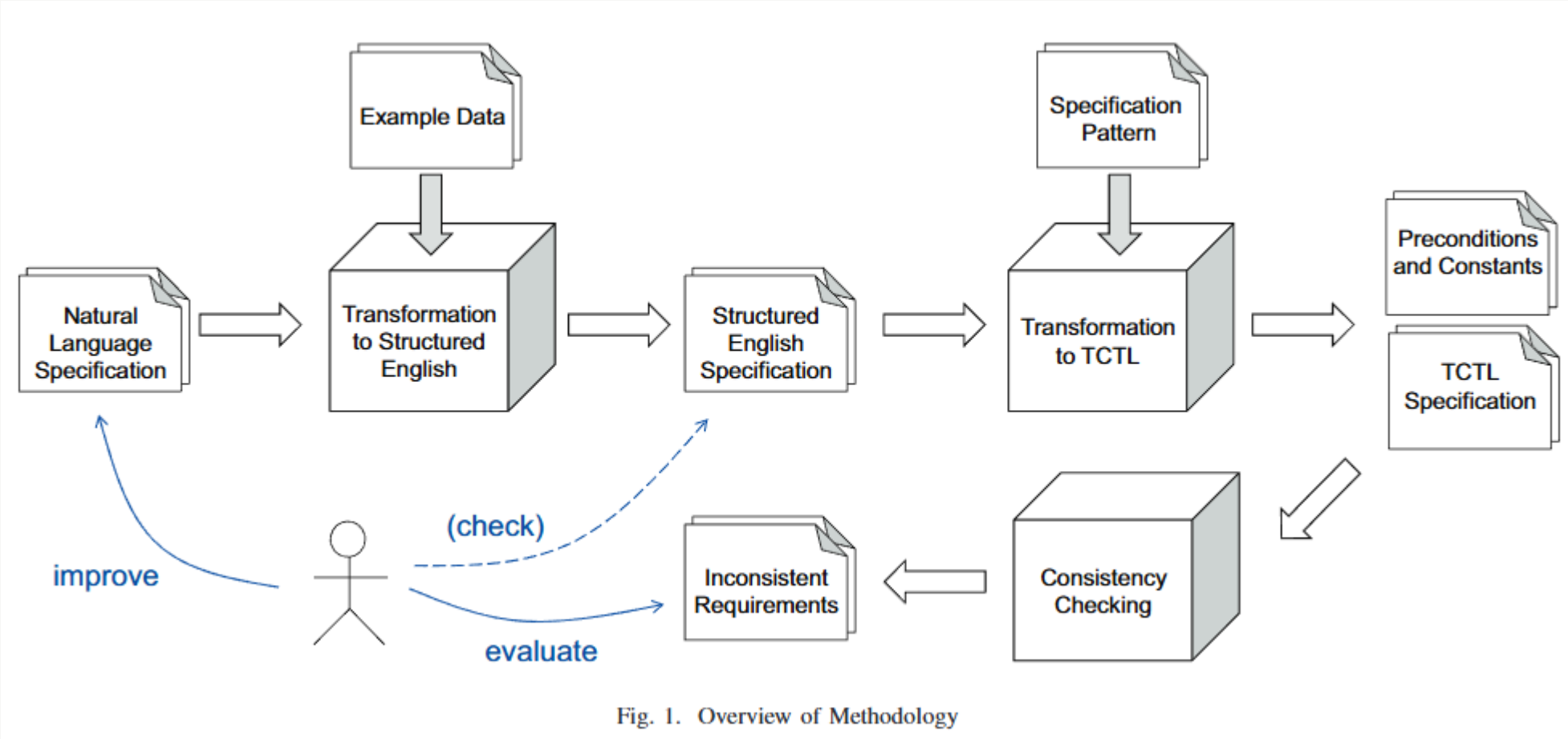


Fig. 1. Overview of Methodology

Bertram, Vincent, et al. "Leveraging natural language processing for a consistency checking toolchain of automotive requirements." ***2023 IEEE 31st International Requirements Engineering Conference (RE)*****. IEEE, 2023.**

Management

AI refines and adapts requirements.

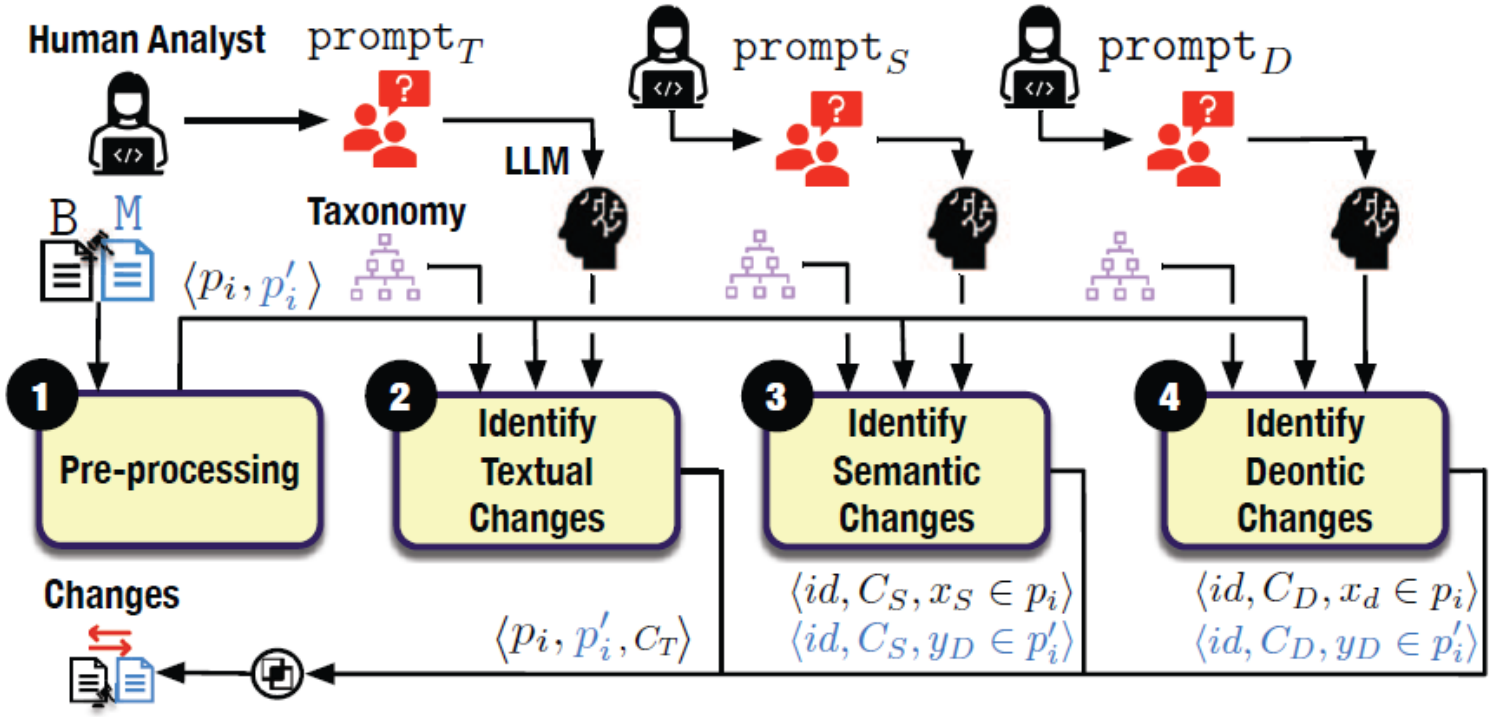


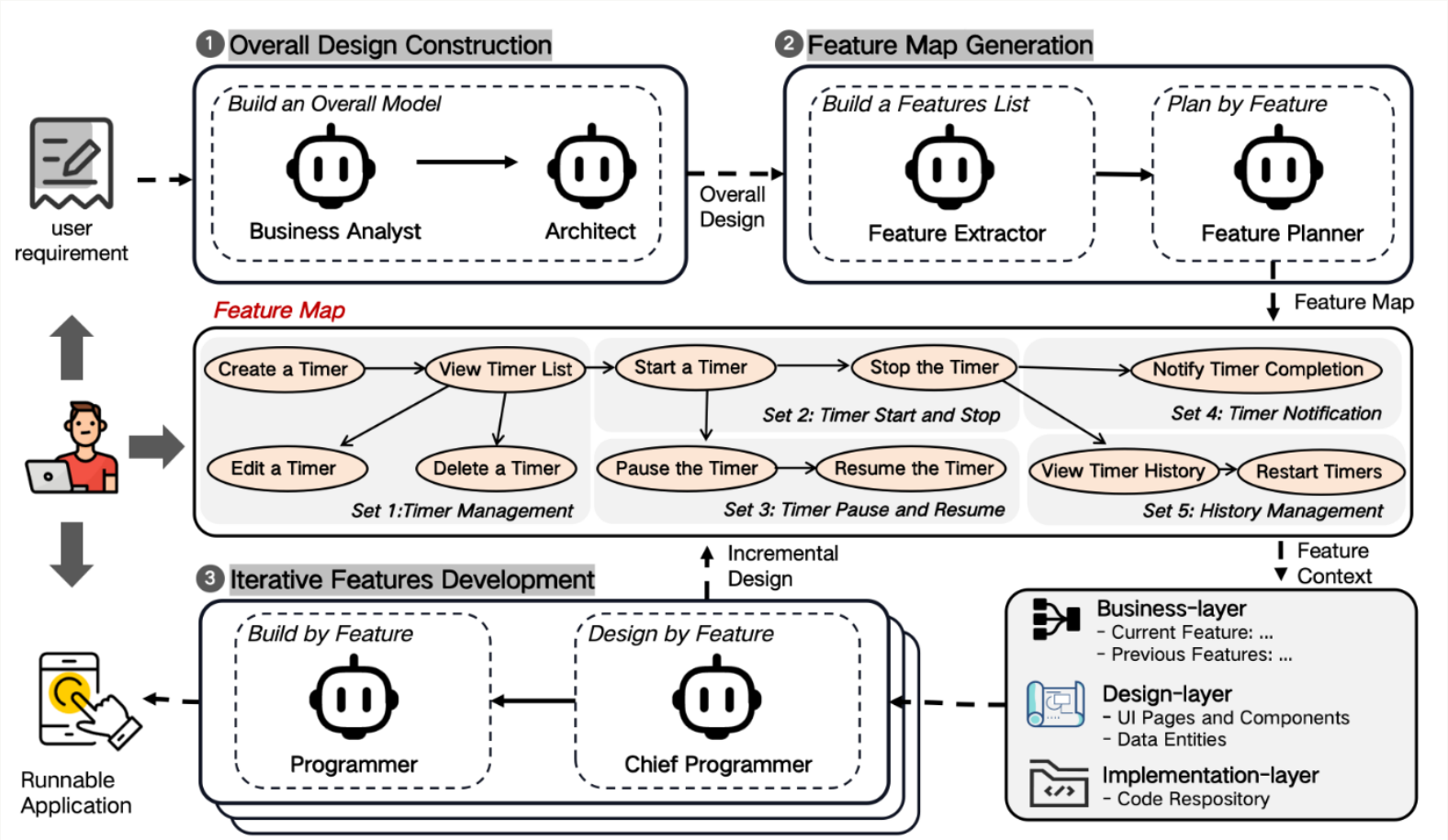
Fig. 3: Overview of MURCIA (B: base act, M: modified act, p_i and p_i': textual strings of provisions in B and M, respectively. C_T, C_S, and C_D: textual, semantic, and deontic changes identified between p_i and p_i', respectively. prompt$_T$, prompt$_S$, and prompt$_D$: prompts to identify textual, semantic, and deontic changes, respectively)

Four-layers change detection for revising specifications.

Abualhaija, Sallam, et al. "AI-enabled regulatory change analysis of legal requirements." ***2024 IEEE 32nd International Requirements Engineering Conference (RE)*****. IEEE, 2024.**

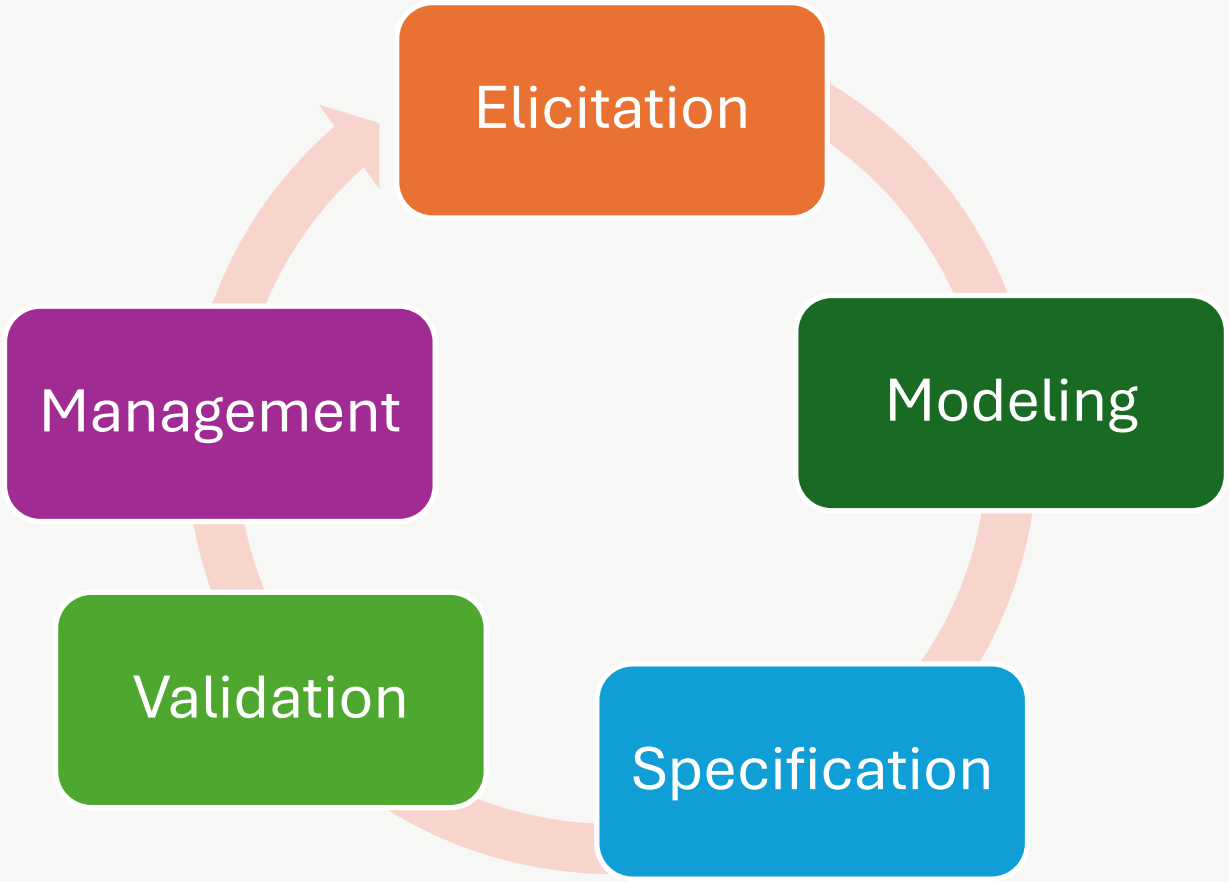
Feature Map

Dependencies across features.



https://arxiv.org/pdf/2511.02399

Requirement Engineering



What are strength and weaknesses?

In AI-assisted software engineering,

Requirements are the most important



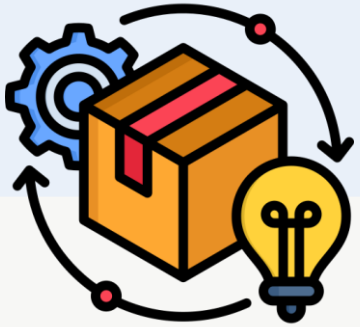
We do not need requirements as code can be directly generated

Strength

Thinking before action



Software engineering life cycle



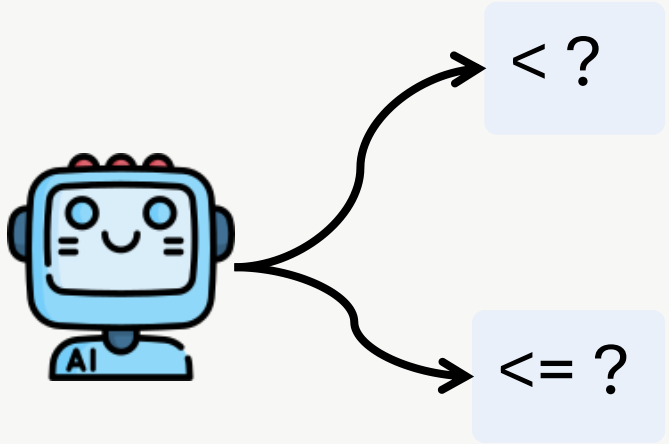
Navigating AI agents



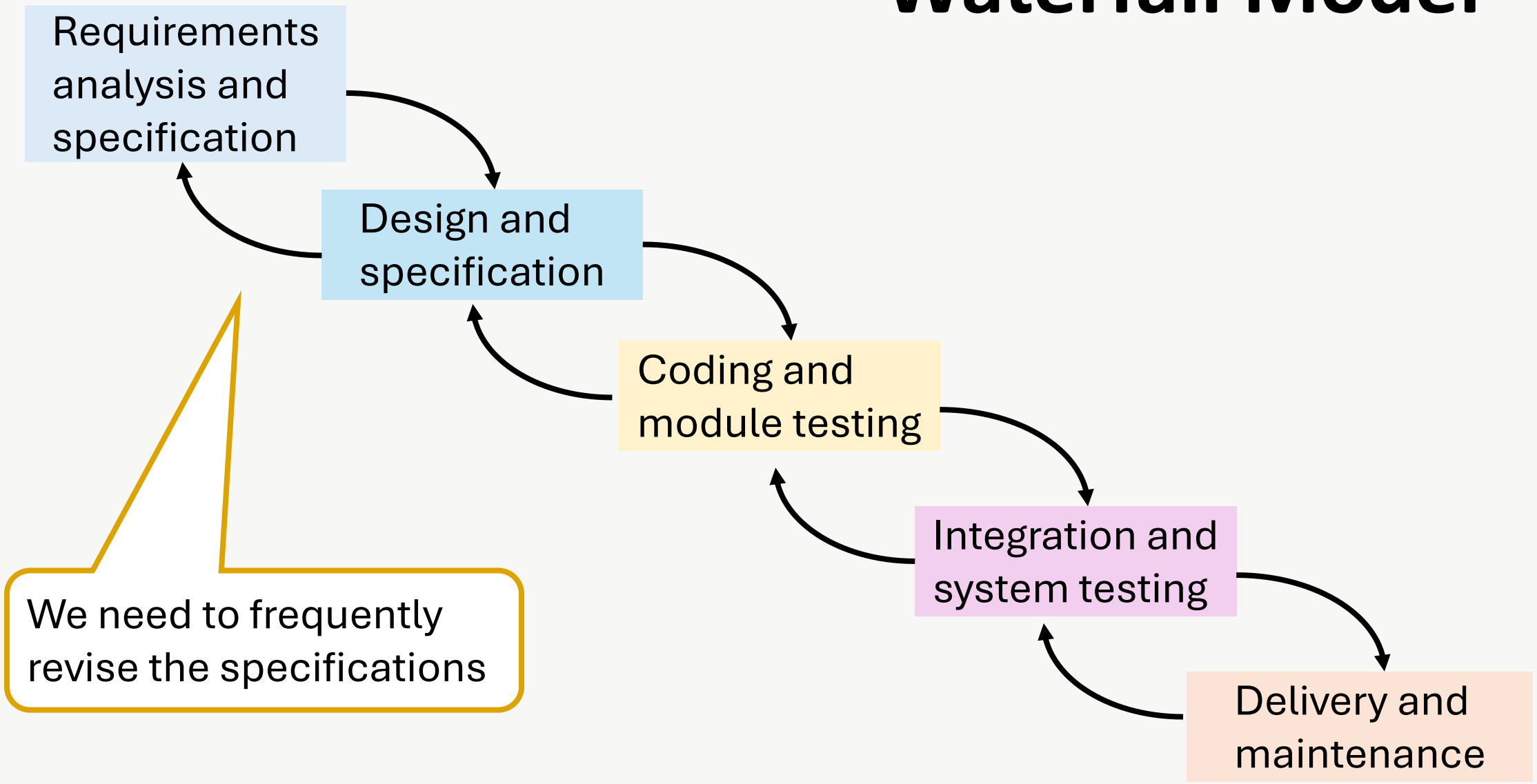
Weakness

- Informal language incurs ambiguity

Insufficient user balance reject transactions.

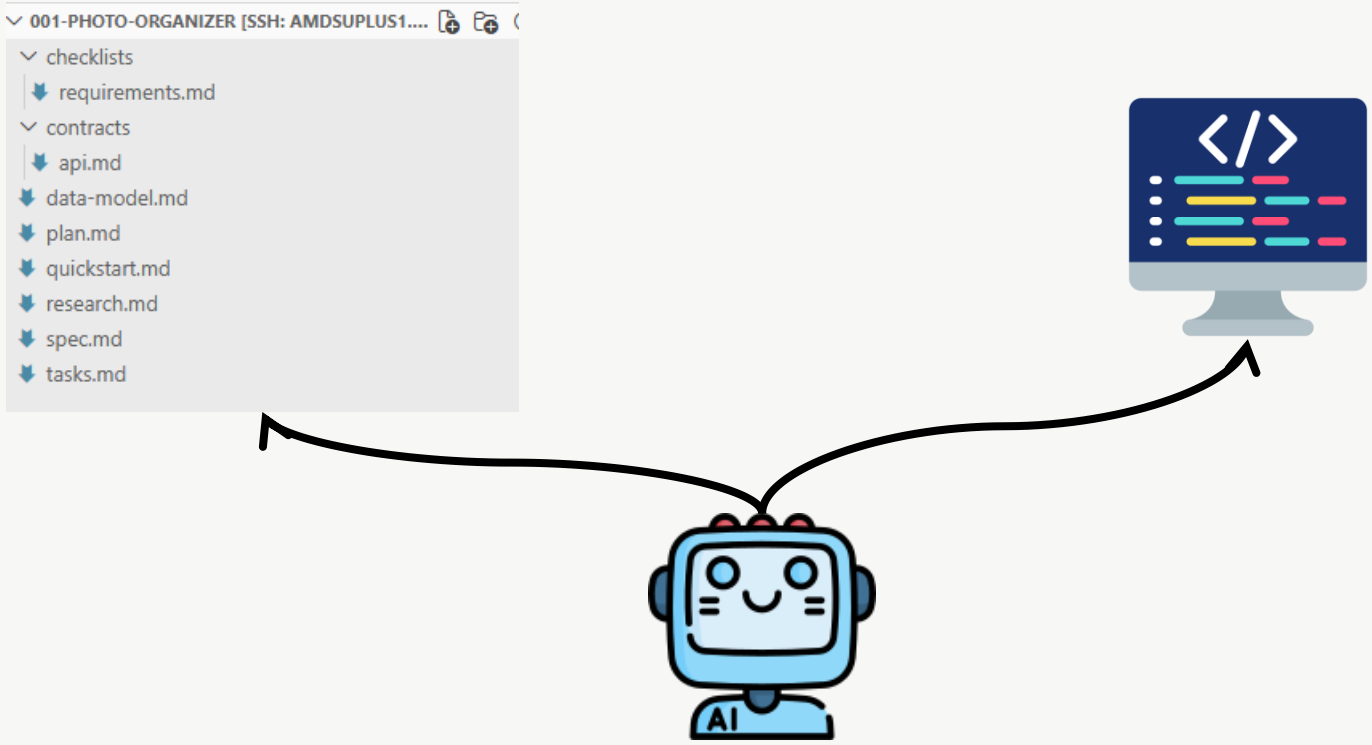


Waterfall Model



Weakness

- Informal language incurs ambiguity
- Much workload of switching writing code and spec





香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Questions

Next class: Requirements Exercise