unibz

Can Test Driven Development be speeded up with Generative AI?

SFSCon 2024

November 8th, 2024

Who are we?

Moritz Mock

Jorge Melegati

Barbara
Russo

(Generative) Large Language Models

- Generative models are types of Machine Learning models (ML) that are designed to produce new data samples that resemble a given dataset
  - For instance, they can predict the next token based on previous tokens, one token at a time

- Example: GPT-4o

Some limitations

- Lack of explainability
- Hallucinations: output that sound plausible but is not true

Explainability

- *Explainability: allows human users to comprehend and trust the results and output created by machine learning algorithms*

- State-of-the-art ML-models tend to be highly complex and black-box
  - GPT-3 has 175 billion parameters!
  - Impossible for humans to reason on these numbers!

Hallucinations

AI hallucinates software packages and devs download them – even if potentially poisoned with malware

- ChatGPT recommends the use of a software library, package, or framework that doesn’t exist
- An attacker can upload a malicious package with the same name to the registries and wait for people to download the packages

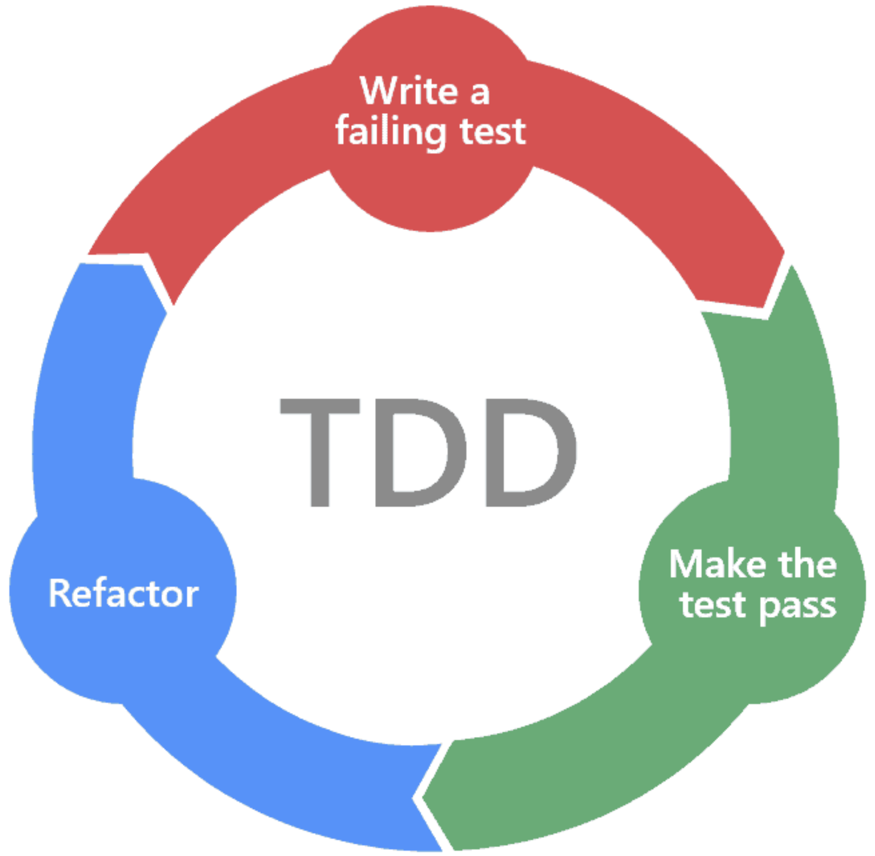
https://www.theregister.com/2024/03/28/ai_bots_hallucinate_software_packages/

AI in software development

- Programming languages are a form of language
- A reasonable use for Generative AI
- How can we tackle the issues mentioned earlier?
  - Test Driven Development can be useful
  - Guaranteeing the existence of tests for the generated code

Test Driven Development

- Writing failing test case
- Minimal code to fulfil the test case
- Refactoring the code



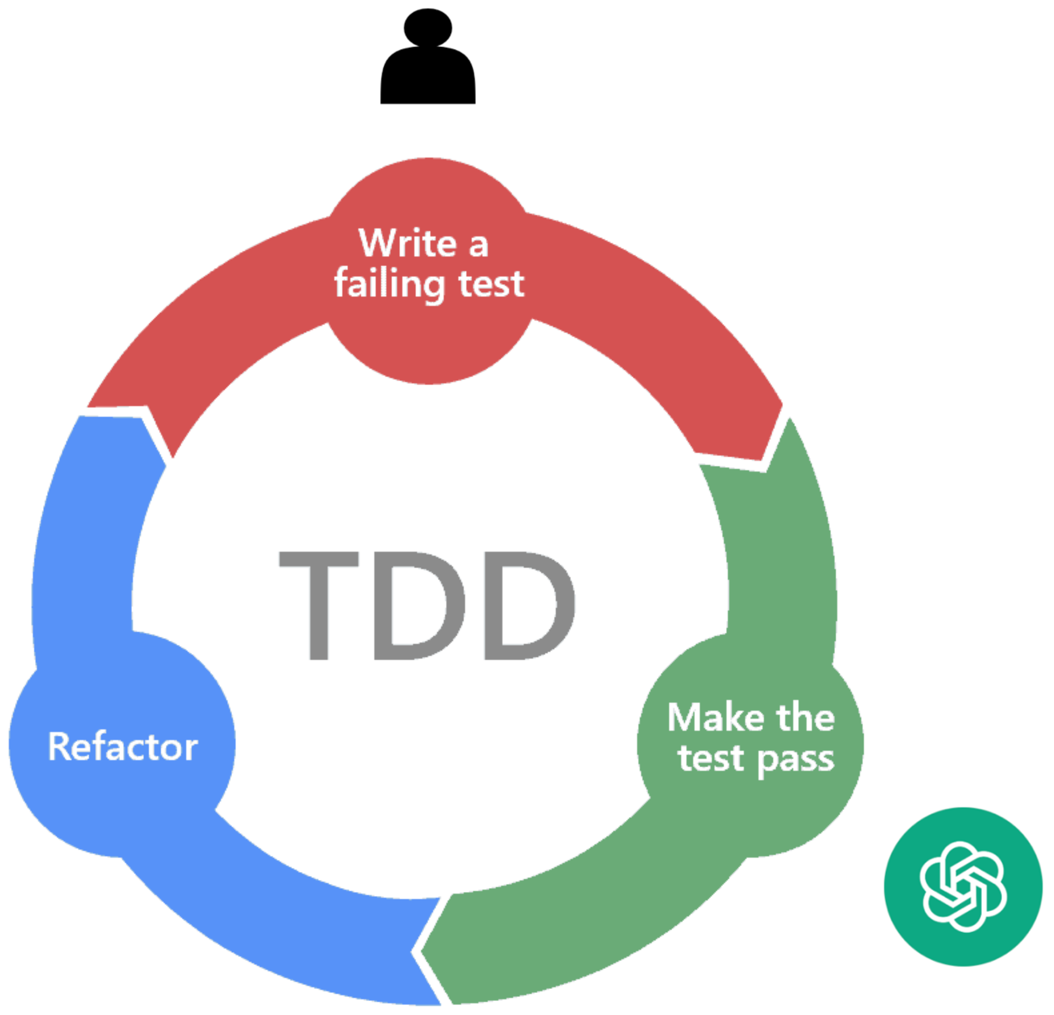
Problem statement

Can generative AI be used to automate TDD?

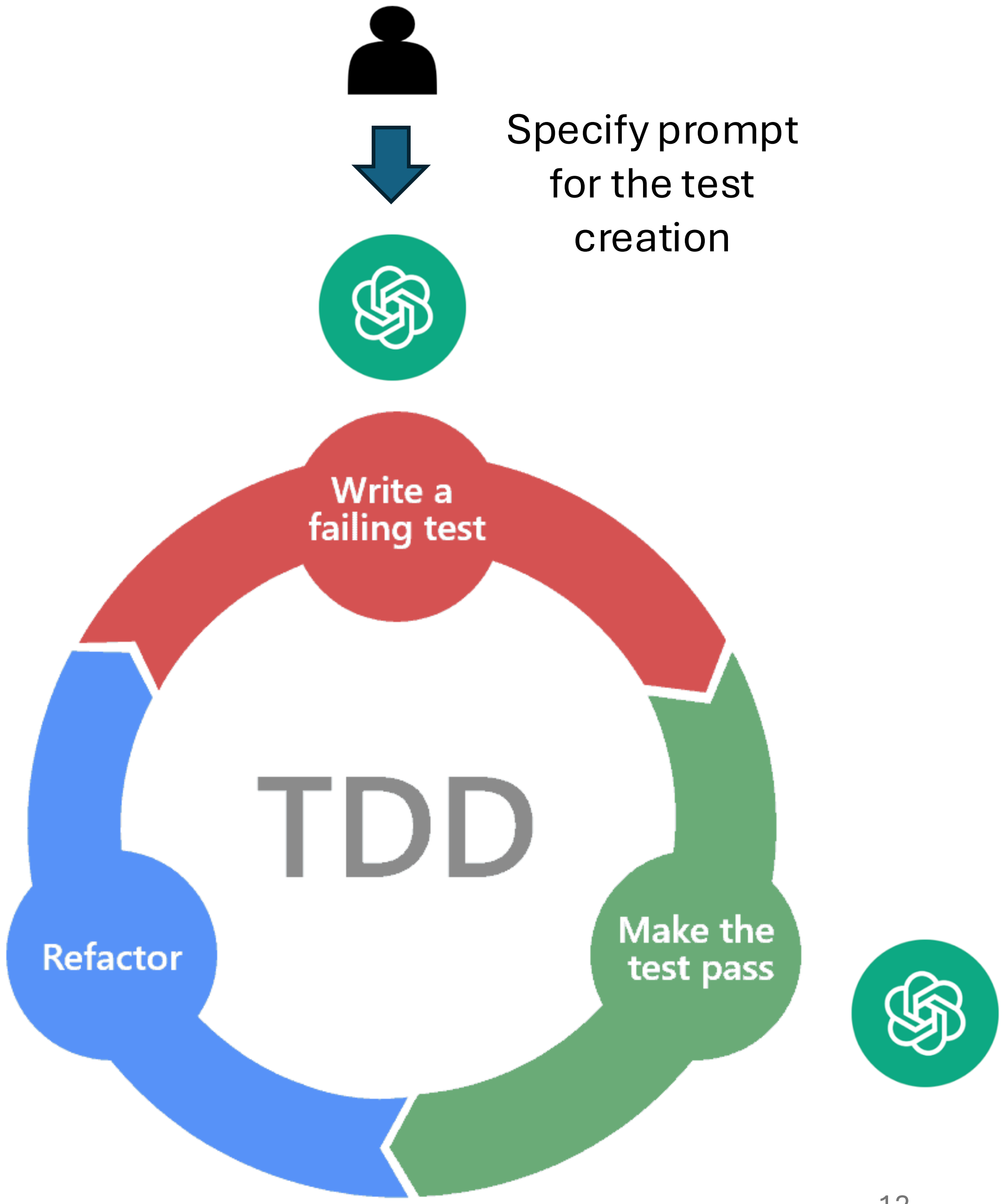
Prompt engineering

- Two interaction patterns
  - Collaborative pattern, one agent
  - Fully-automated pattern, two agents
- Created dedicated prompts
- Used ChatGPT’s API
- For each interaction a new agent was considered

Integration of AI in TDD Collaborative pattern



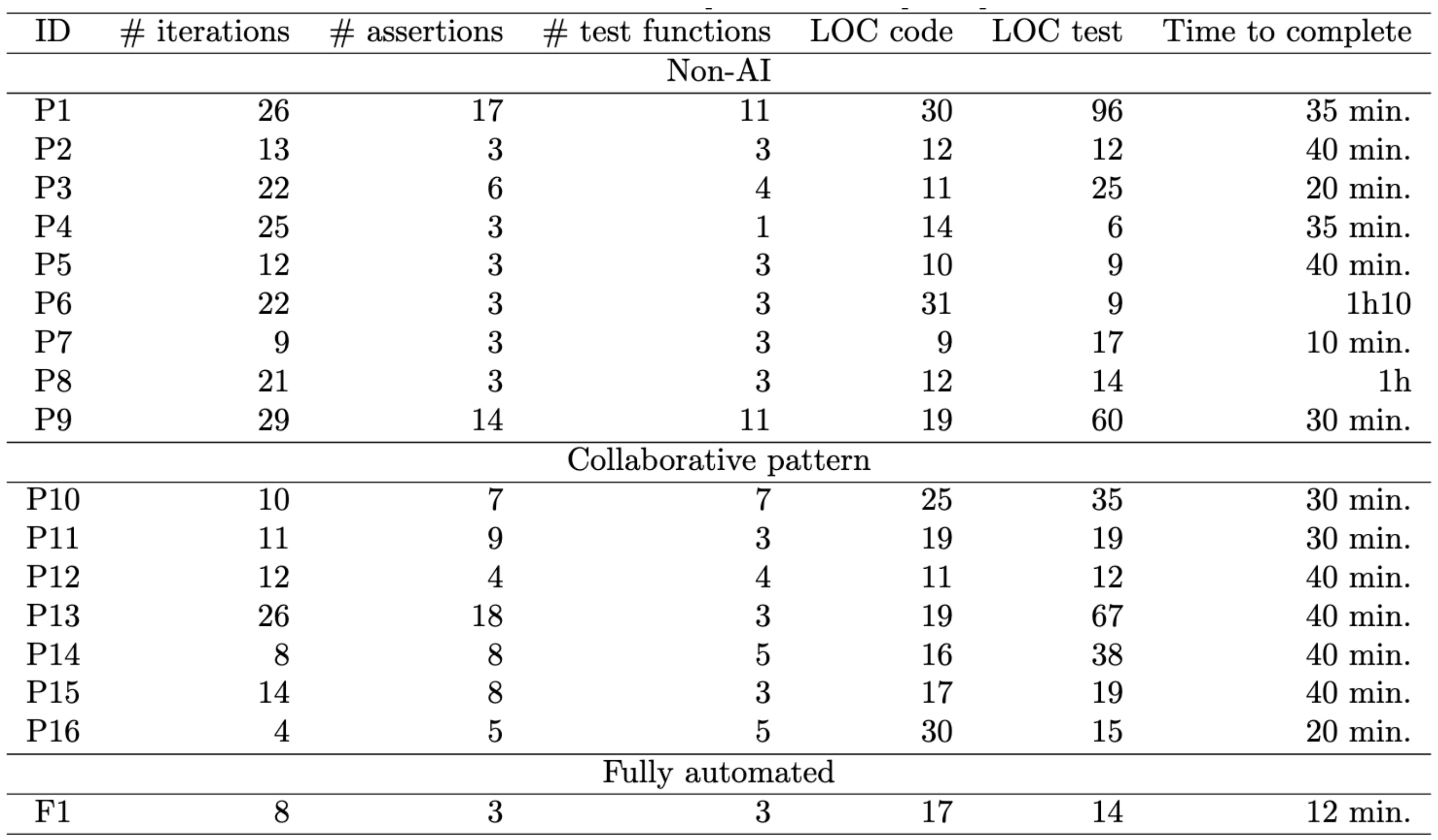
Integration of AI in TDD
Fully automated pattern



Experiment setup

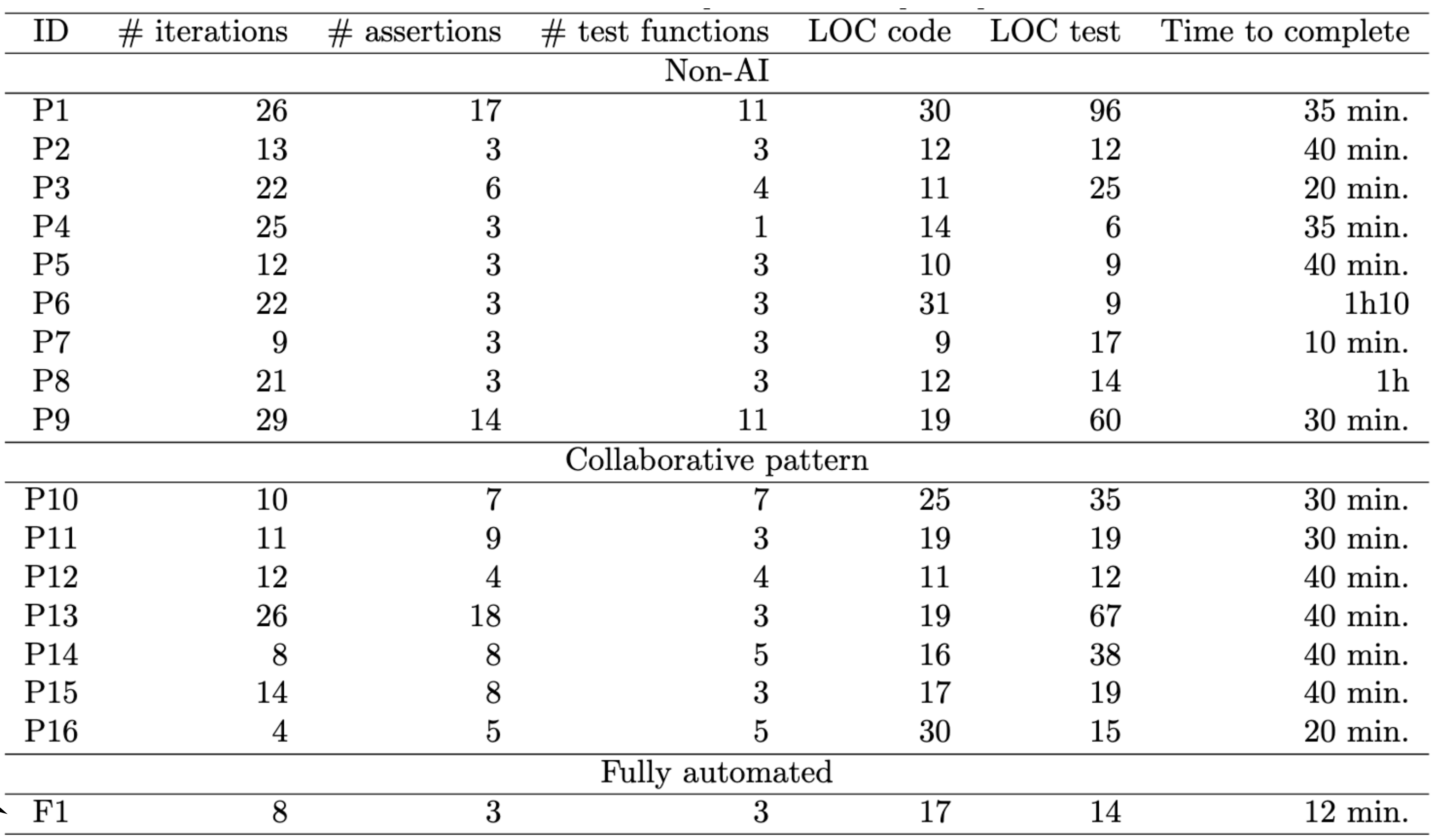
- Non-automated TDD vs. Collaborative pattern
- Experiment in Python
- Conducted online (Google Colab)
- 40 minutes to complete the exercise
- Employed the gpt-3.5-turbo model

Results of the experiment



ID	# iterations	# assertions	# test functions	LOC code	LOC test	Time to complete
			Non-AI			
P1	26	17	11	30	96	35 min.
P2	13	3	3	12	12	40 min.
P3	22	6	4	11	25	20 min.
P4	25	3	1	14	6	35 min.
P5	12	3	3	10	9	40 min.
P6	22	3	3	31	9	1h10
P7	9	3	3	9	17	10 min.
P8	21	3	3	12	14	1h
P9	29	14	11	19	60	30 min.
			Collaborative pattern			
P10	10	7	7	25	35	30 min.
P11	11	9	3	19	19	30 min.
P12	12	4	4	11	12	40 min.
P13	26	18	3	19	67	40 min.
P14	8	8	5	16	38	40 min.
P15	14	8	3	17	19	40 min.
P16	4	5	5	30	15	20 min.
			Fully automated			
F1	8	3	3	17	14	12 min.

Results of the experiment

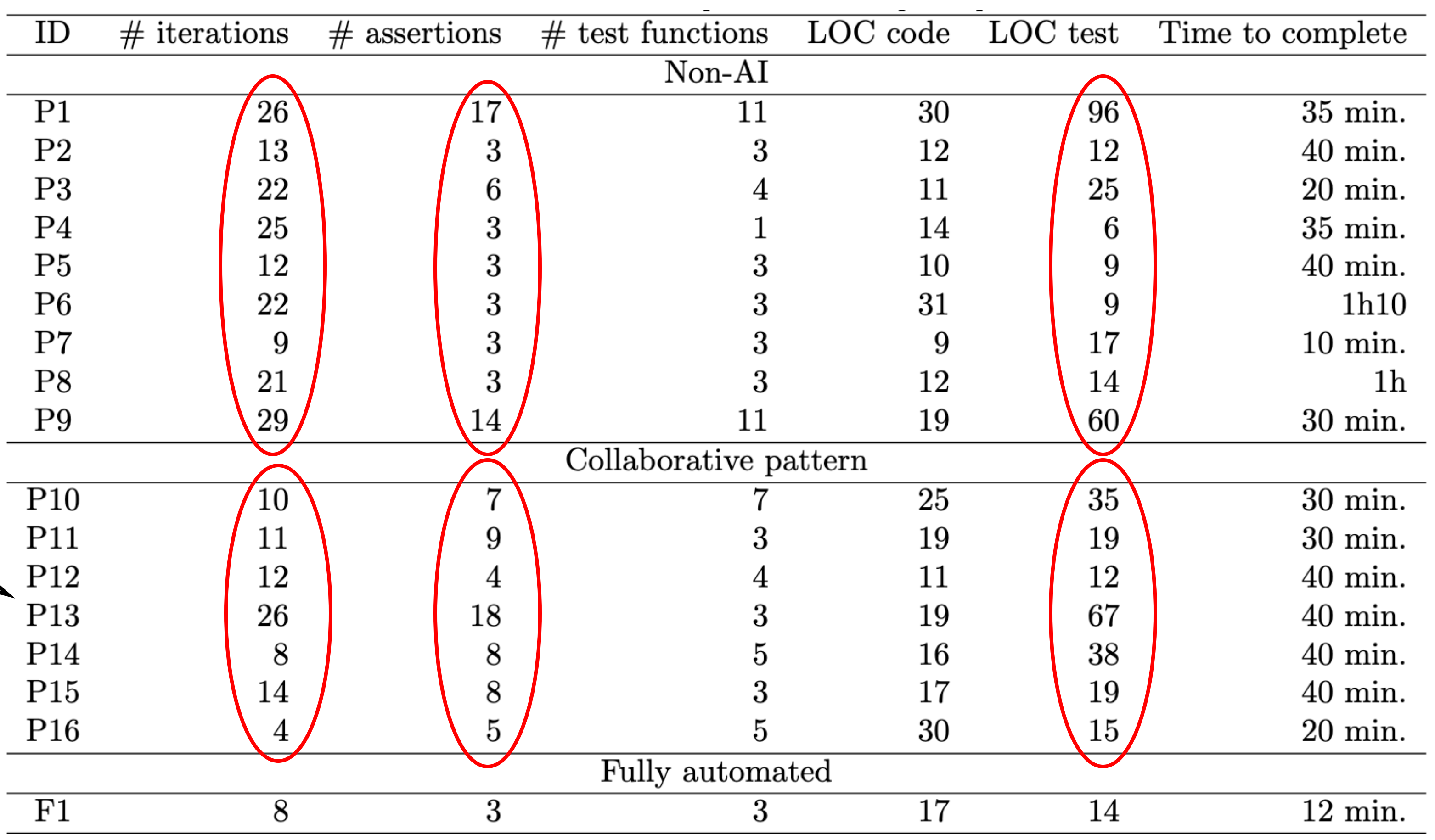


ID	# iterations	# assertions	# test functions	LOC code	LOC test	Time to complete
			Non-AI			
P1	26	17	11	30	96	35 min.
P2	13	3	3	12	12	40 min.
P3	22	6	4	11	25	20 min.
P4	25	3	1	14	6	35 min.
P5	12	3	3	10	9	40 min.
P6	22	3	3	31	9	1h10
P7	9	3	3	9	17	10 min.
P8	21	3	3	12	14	1h
P9	29	14	11	19	60	30 min.
			Collaborative pattern			
P10	10	7	7	25	35	30 min.
P11	11	9	3	19	19	30 min.
P12	12	4	4	11	12	40 min.
P13	26	18	3	19	67	40 min.
P14	8	8	5	16	38	40 min.
P15	14	8	3	17	19	40 min.
P16	4	5	5	30	15	20 min.
			Fully automated			
F1	8	3	3	17	14	12 min.

Fully automated: fast and accurate but no tests for edge cases.

Results of the experiment

Collaborative pattern: less interactions but increased number and size of tests.



ID	# iterations	# assertions	# test functions	LOC code	LOC test	Time to complete
			Non-AI			
P1	26	17	11	30	96	35 min.
P2	13	3	3	12	12	40 min.
P3	22	6	4	11	25	20 min.
P4	25	3	1	14	6	35 min.
P5	12	3	3	10	9	40 min.
P6	22	3	3	31	9	1h10
P7	9	3	3	9	17	10 min.
P8	21	3	3	12	14	1h
P9	29	14	11	19	60	30 min.
			Collaborative pattern			
P10	10	7	7	25	35	30 min.
P11	11	9	3	19	19	30 min.
P12	12	4	4	11	12	40 min.
P13	26	18	3	19	67	40 min.
P14	8	8	5	16	38	40 min.
P15	14	8	3	17	19	40 min.
P16	4	5	5	30	15	20 min.
			Fully automated			
F1	8	3	3	17	14	12 min.

Conclusion

- For our experimental settings, generative AI can be used to automate TDD
- With the abstraction of the human in the TDD process the level of creativity may get worse
- The AI needs expert supervision
    - A junior developer might be misled by the AI-generated solution

Thank you!