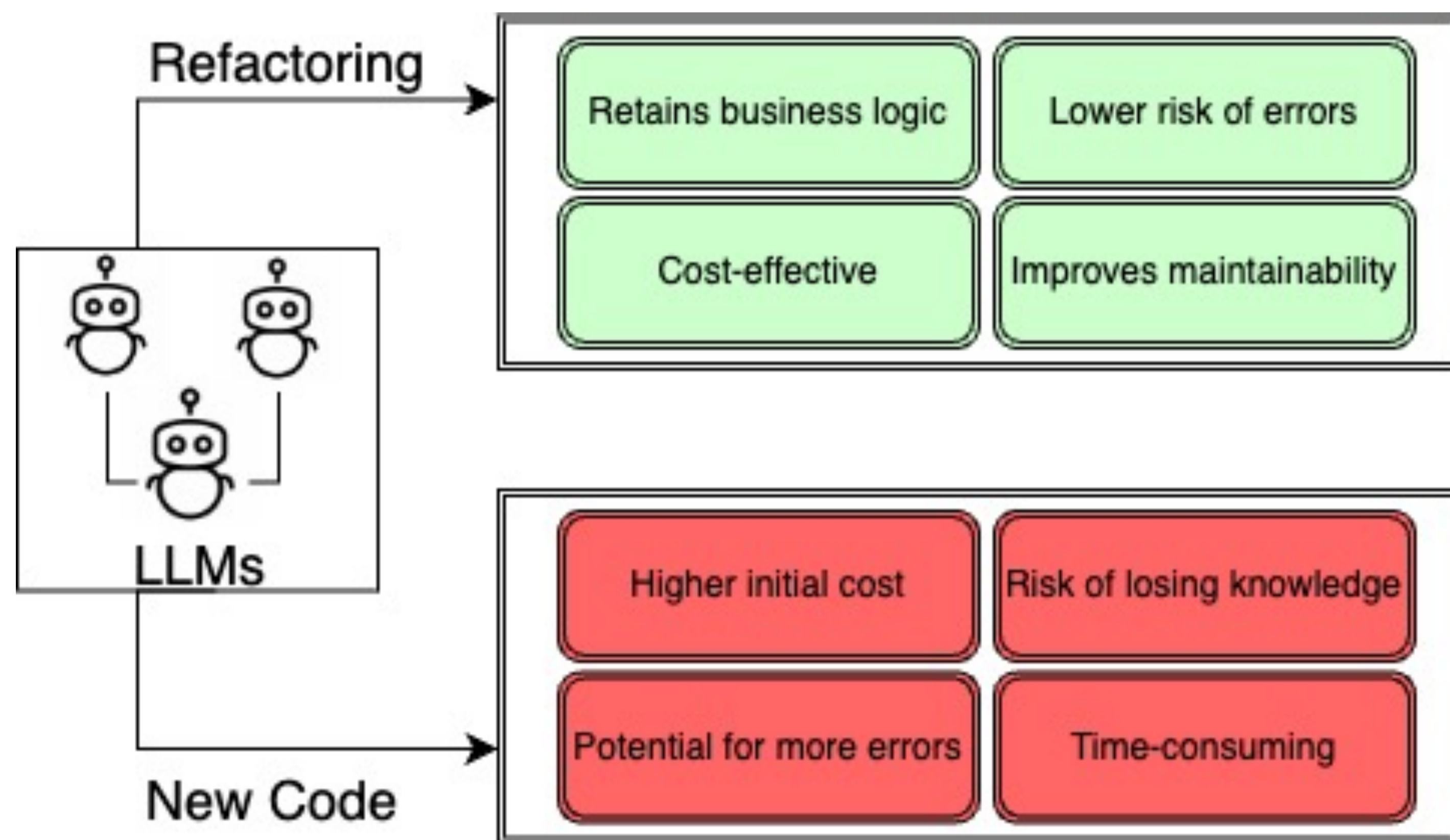


# Multi-Agent Refactoring in Functional Languages for Empowered Software Development.

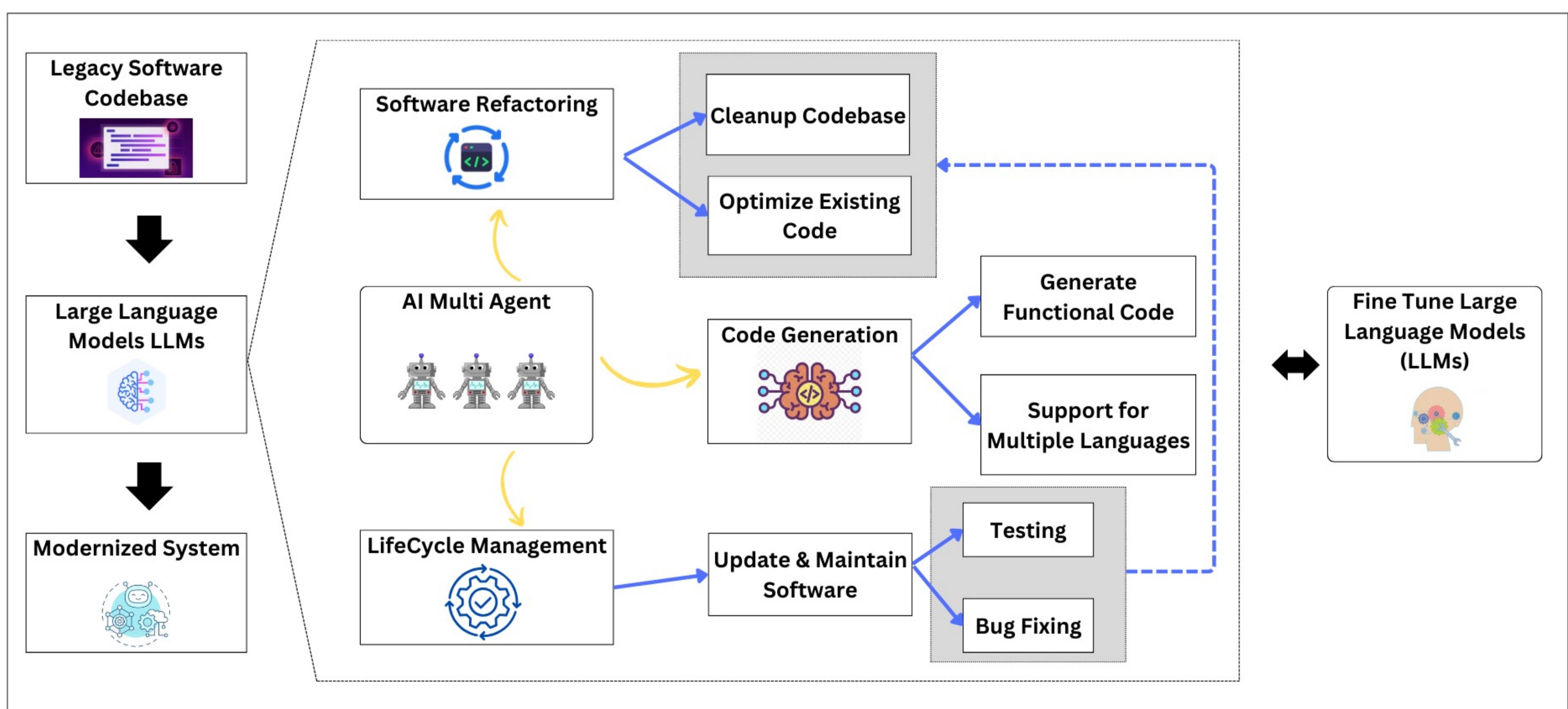
## Research Questions



**RQ1** - How do LLMs make refactoring existing software more advantageous than starting development from scratch?

**RQ2** - How effective are LLMs in bridging the gap between functional languages like Haskell and other programming paradigms?

**RQ3** - How can LLMs be leveraged to support the full software lifecycle?



## Challenges

Contextual Understanding and Consistency<sup>[1][2]</sup>

Handling Functional Programming Paradigms<sup>[2]</sup>

Code Complexity and Scalability<sup>[1]</sup>

Testing and Verification<sup>[1]</sup>

Domain-Specific Knowledge<sup>[2]</sup>

## Conclusion

The integration of Large Language Models (LLMs) into software development processes holds great promise for improving efficiency and reducing the cognitive load on developers. This research aims to provide a comprehensive understanding of the potential benefits and challenges associated with using LLMs for software refactoring, generation, and lifecycle management. By exploring these questions, we hope to pave the way for more intelligent, automated, and reliable software development practices.

## References

- Choi, J., An, G. and Yoo, S., 2024, July. Iterative Refactoring of Real-World Open-Source Programs with Large Language Models. In International Symposium on Search Based Software Engineering (pp. 49-55). Cham Springer Nature Switzerland.
- Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R. and McHardy, R., 2023. Challenges and applications of large language models. arXiv preprint arXiv:2307.10169.

**Supervisor:** Professor Pekka Abrahamsson  
[pekka.abrahamsson@tuni.fi](mailto:pekka.abrahamsson@tuni.fi)

<https://gpt-lab.eu>



More Info