

# RefactorAI

## AI-based Refactoring of Legacy Systems for Digital Transformation Use Case – Support of Instrumentation Upgrade



Pro<sup>2</sup>Future

Lukas Freiseisen<sup>2</sup>, Claudio Di Sipio<sup>2</sup>, Nour Oulad Moussa<sup>1</sup>, Dmitrii Sharshakov<sup>2</sup>, Oujidane Guiza<sup>1</sup>, Alexander Egyed<sup>2</sup>, Paul Grünbacher<sup>2</sup>, Markus Jäger<sup>1</sup>, Andreas Hametner<sup>3</sup>, Moritz Witt<sup>3</sup>, Rainer Klaffenböck<sup>3</sup>

Pro2Future GmbH<sup>1</sup>, JKU-ISSE, Dynatrace Austria GmbH<sup>3</sup>

<sup>1</sup> Science Park 4, Altenberger Strasse 69, 4040 Linz

<sup>2</sup> Science Park 3, Altenberger Strasse 69, 4040 Linz

<sup>3</sup> Am Fünfundzwanziger Turm 20, 4020 Linz



### MOTIVATION & GOALS

Complex large-scale software systems or systems of systems rely on **Software Tracing** to understand their behaviour and to stay maintainable. Systems are thus **instrumented** to produce telemetry data during runtime. Nowadays, systems are engineered within **rapidly changing ecosystems of programming languages** they are based on. **All these libraries need to be supported by tracing frameworks.** This is where our use case comes into play.

RefactorAI aims to combine **deterministic algorithms** with **Large Language Models** to build an **agentic system** that helps developers to **refactor existing instrumentations** to be applicable to **new target library releases**.

#### Project FactBox

**Project Name** RefactorAI  
**Project ID** FFG 921370  
**Duration** 36 Months

**Area 2**  
Area Orchestration

**Project Lead**  
Dr. Oujidane Guiza

### APPROACH

When trying to solve problems with Large Language Models, the biggest challenge is to gather and provide the necessary data and to build a context harness that minimizes non-determinism.

RefactorAI aims to address these issues by collecting and processing source code, diffs, execution traces and other meta-data. This information is then presented to specialized agents with limited capabilities to minimize hallucinations.

### CONTRIBUTION

#### Scientific contribution

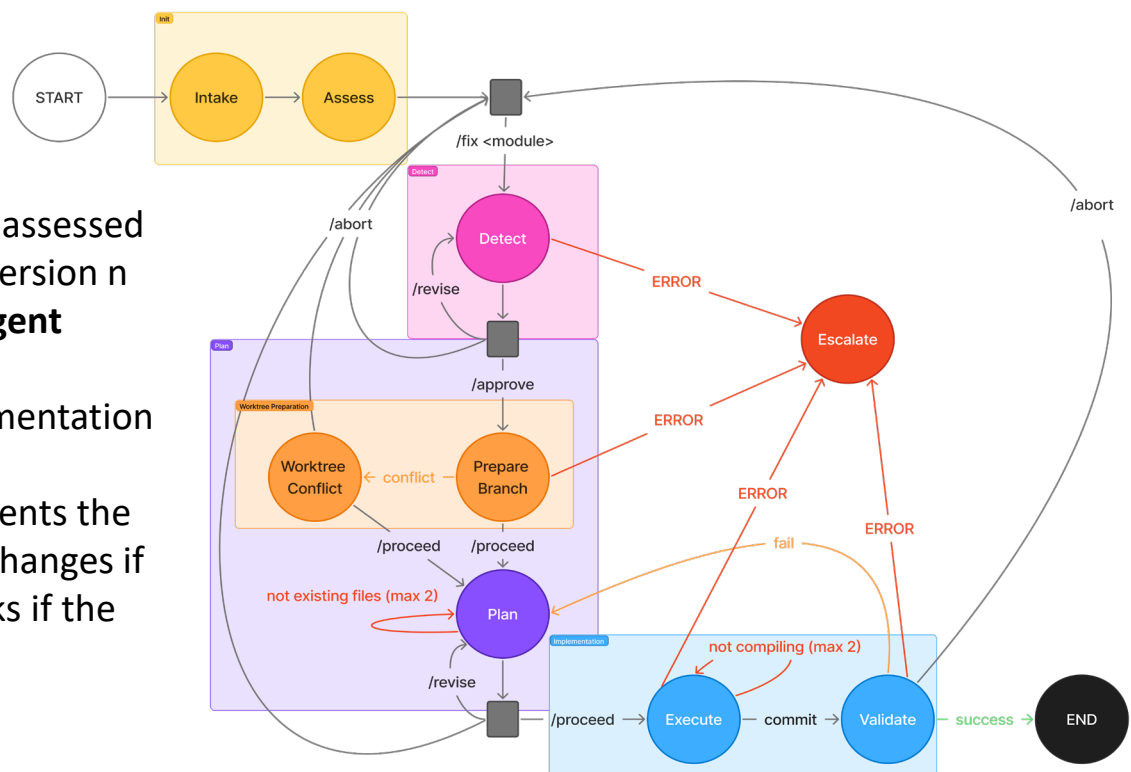
Combination of static/dynamic analysis of source code and execution graphs to build a context harness for a multi-agent system that solves the refactoring task.

#### Economic contribution

Assist developers in their every-day work with complex tasks and fully automate trivial and simple cases.

### PROTOTYPE

- **Init:** Preprocessed information is loaded and assessed
- **Detect:** Discrepancies between test runs of version n and version n+1 are detected by a **Detector Agent**
- **Plan:** A **Planner Agent** digests the previously accumulated information and drafts an implementation plan to refactor the instrumentation.
- **Implementation:** An **Executor Agent** implements the according to the given plan and commits the changes if the code is compiling. A **Validator Agent** checks if the implementation works.



**Contact:** Dr. Markus Jäger, Pro2Future GmbH, markus.jaeger@pro2future.at, +43 732 2468 - 4191

**Acknowledgement:** This work was supported by Pro2Future II (FFG, 911655) and Refactor AI (FFG, 921370).

**IT | PRO** **dynatrace**  
Software die's einfach macht.