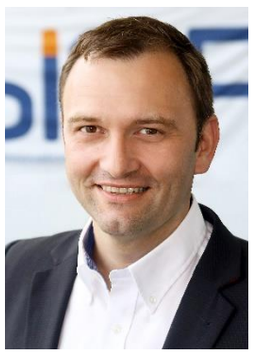
Software Innovation Campus Paderborn

Test Case Co-Migration Method Patterns

Ivan Jovanovikj, Enes Yigitbas, Stefan Sauer, Gregor Engels



25.02.2020

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

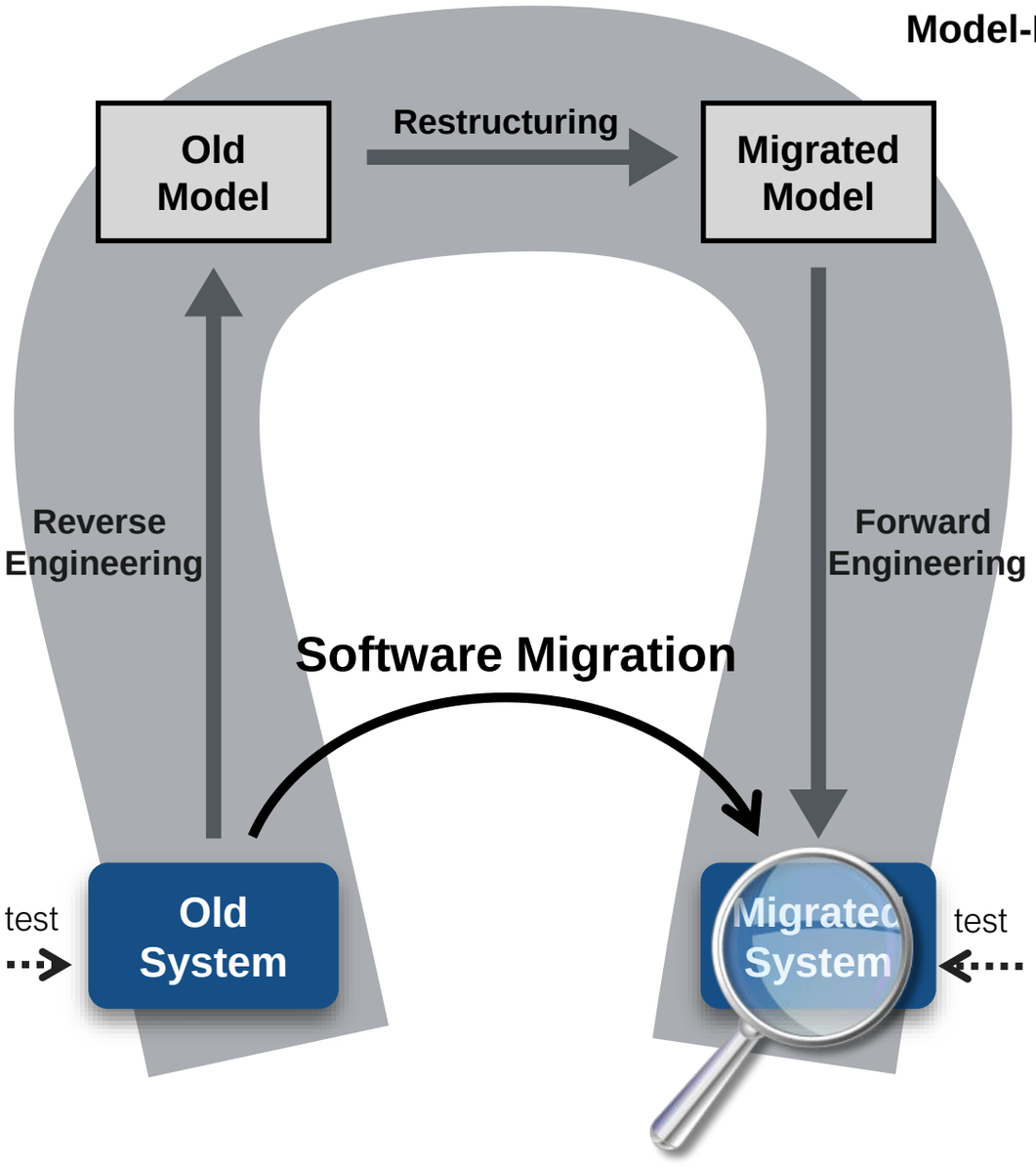
Agenda

1. Reuse of Test Cases in Migration Projects
2. Test Case Co-Migration Method Patterns
3. Overview and Conclusion

Reuse of Test Cases in Migration Projects

Software Innovation Campus Paderborn

Model-Driven Software Migration OMG ADM



Problem Statement

How to systematically reuse existing test cases in a software migration scenario?

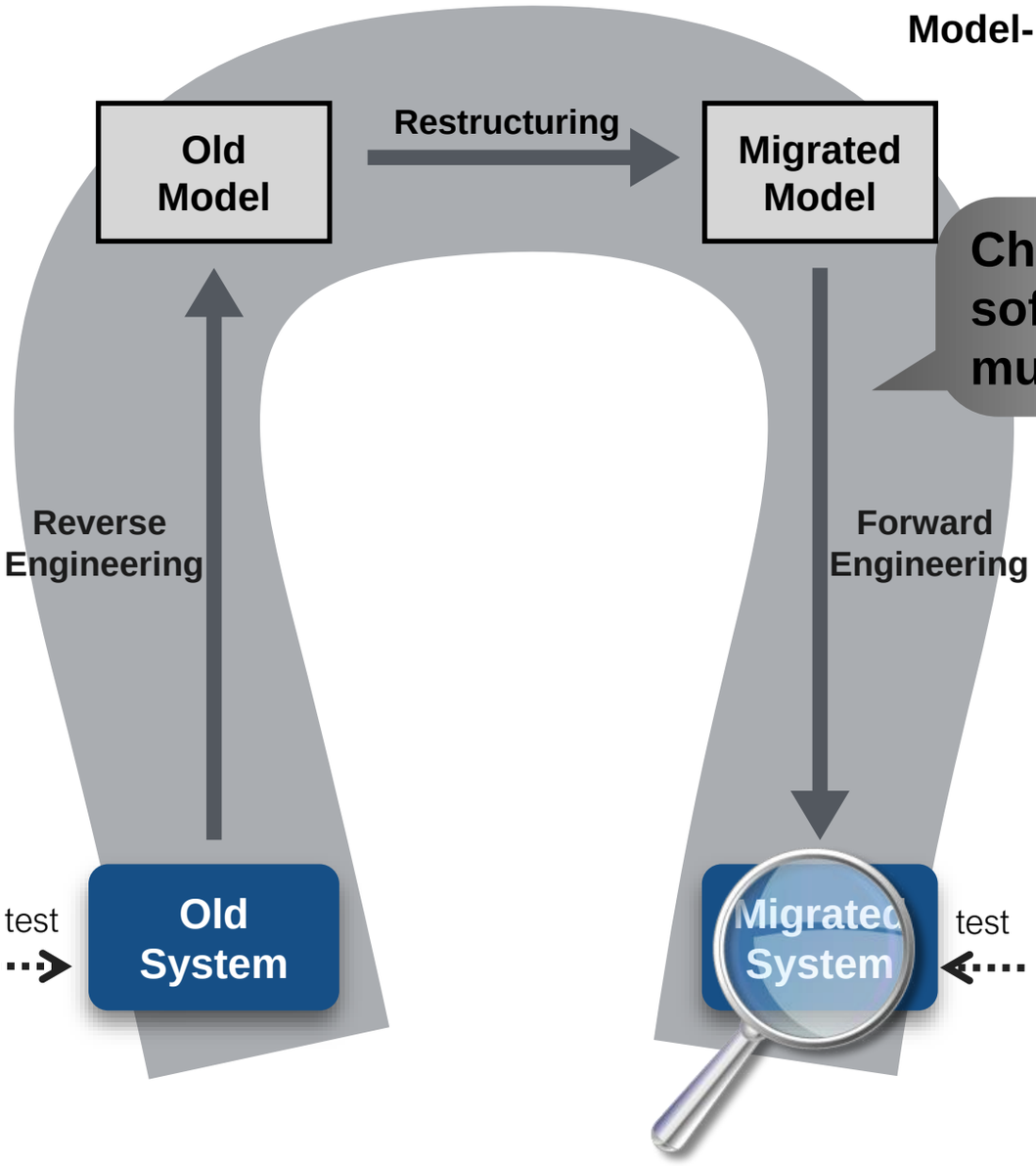
Test the desired functionality of the source system

Test Case Design is costly and time consuming

Reuse of Test Cases in Migration Projects

Software Innovation Campus Paderborn

Model-Driven Software Migration OMG ADM



Problem Statement
How to systematically reuse existing test cases in a software migration scenario?

Changes from the software migration must be considered — **R1**

Test requirements must be considered — **R2**

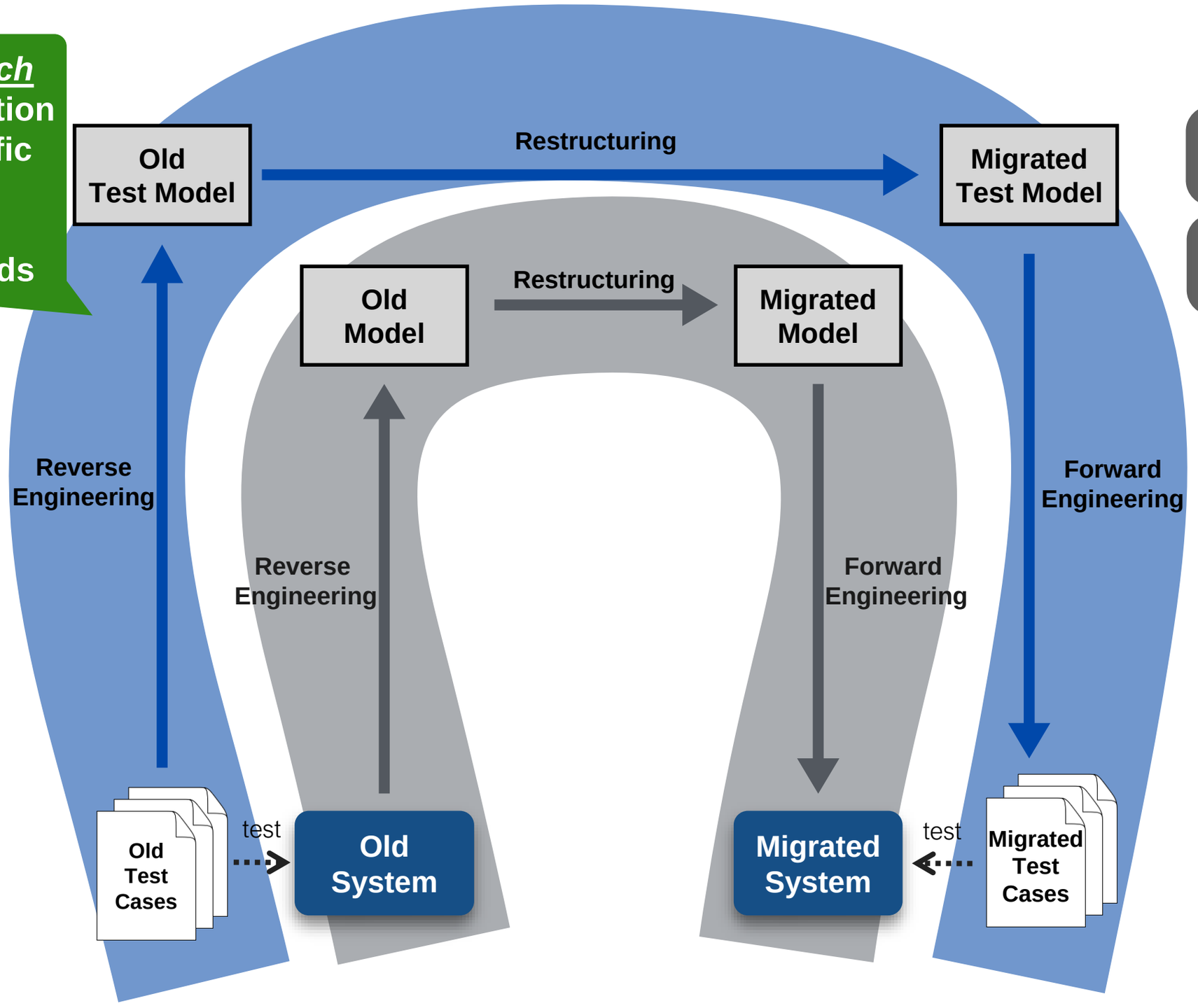
Test the desired functionality of the source system

Test Case Design is costly and time consuming

Reuse of Test Cases in Migration Projects

Software Innovation Campus Paderborn

Solution Approach
Modular construction of context-specific model-driven test case migration methods

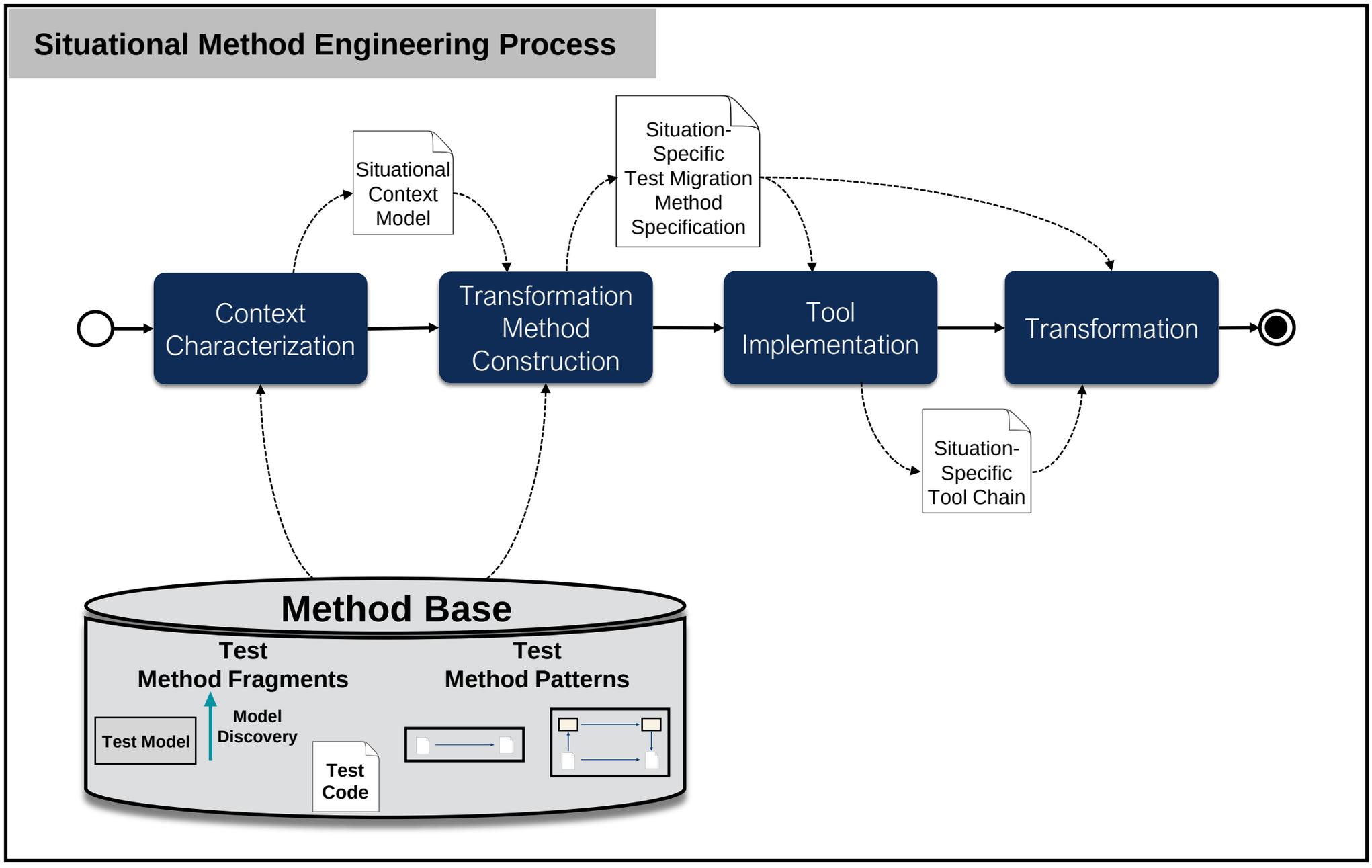


- Co-evolution support — R1
- Context consideration — R2

Agenda

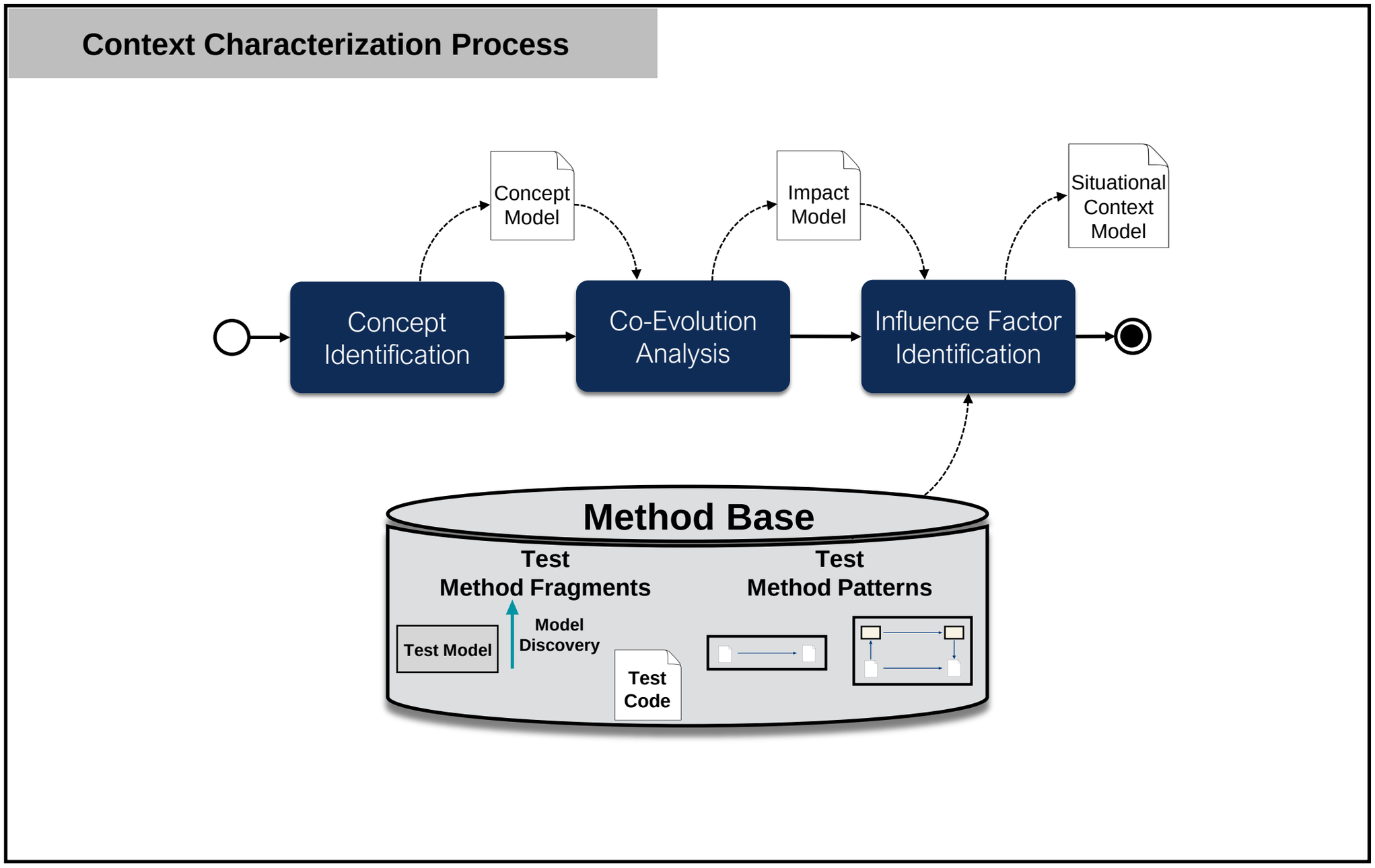
1. **Reuse of Test Cases in Migration Projects**
2. **Test Case Co-Migration Method Patterns**
3. **Overview and Conclusion**

Situation-specific Test Case Co-Migration



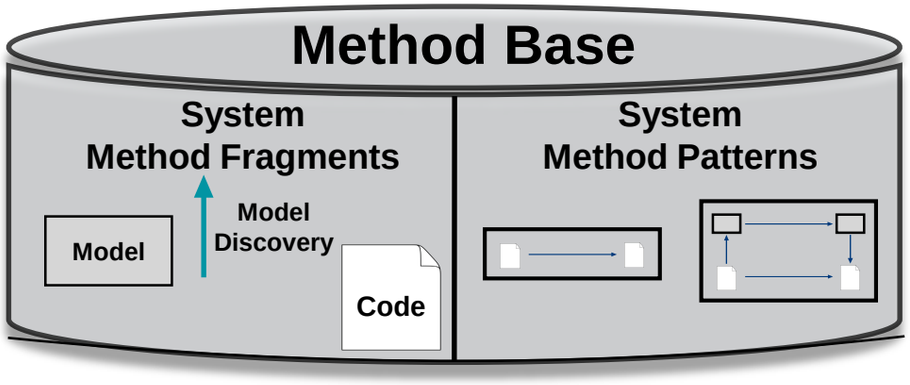
Context Characterization

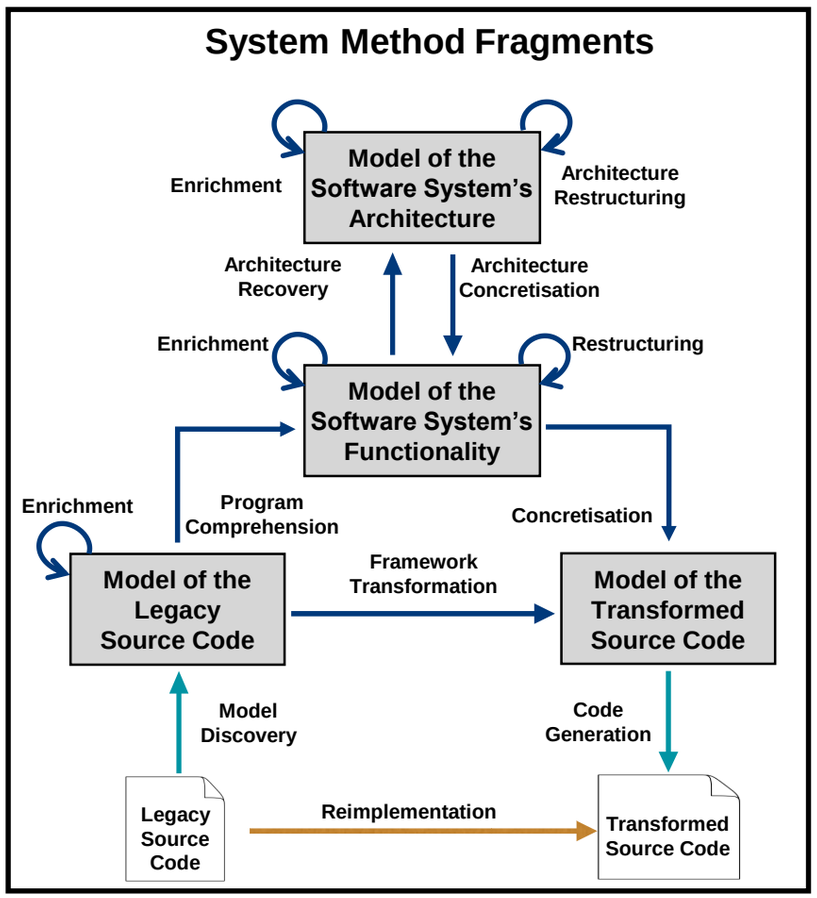
Software Innovation Campus Paderborn

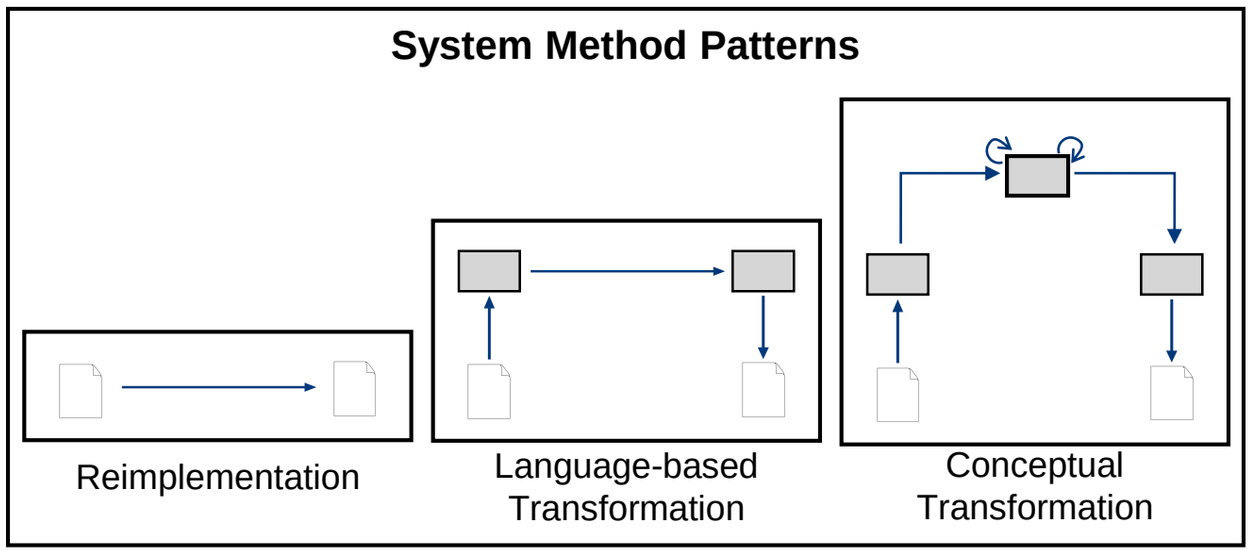


Method Base – System Method Fragments

Software Innovation Campus Paderborn

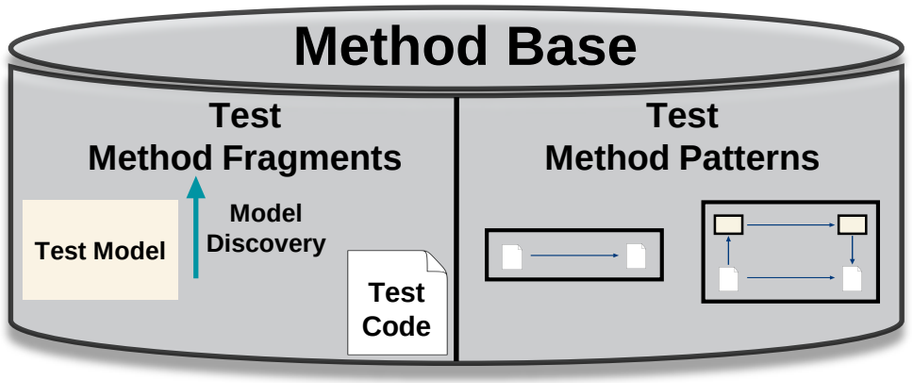


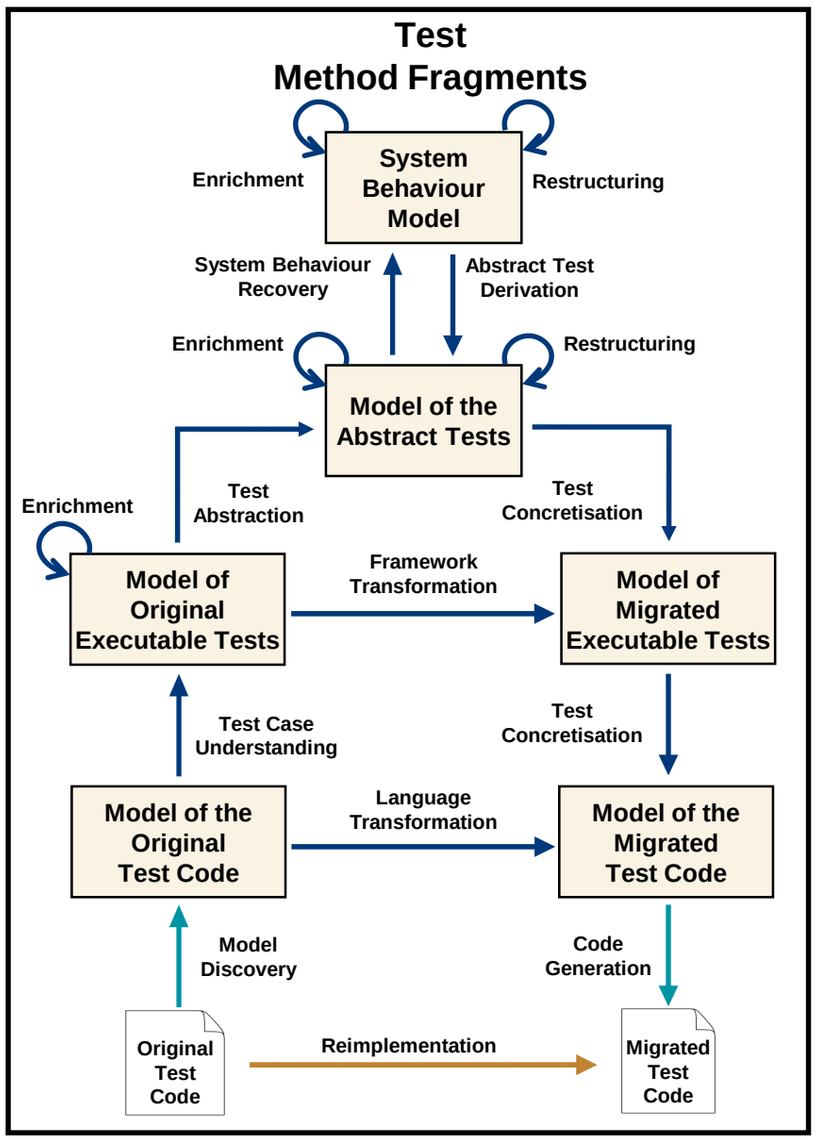


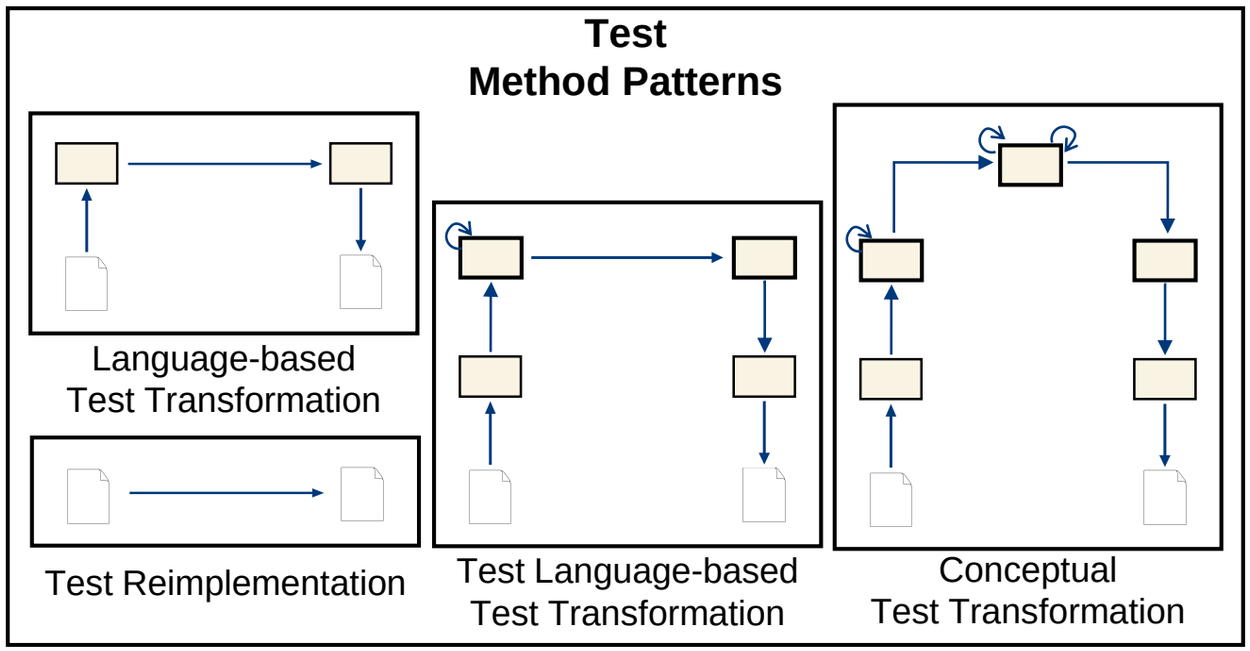


Method Base – Test Method Fragments

Software Innovation Campus Paderborn



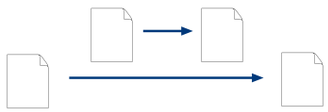
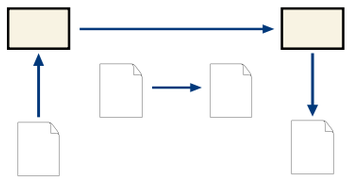
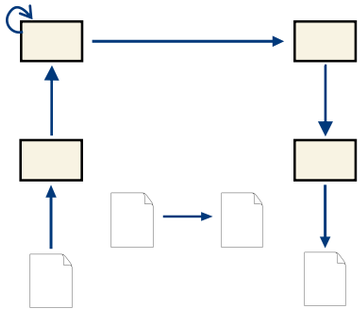
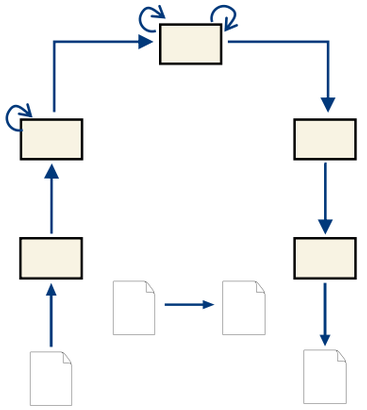
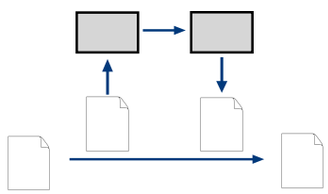
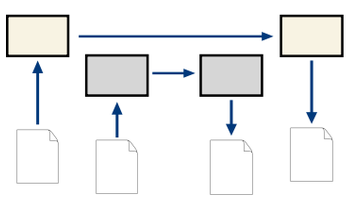
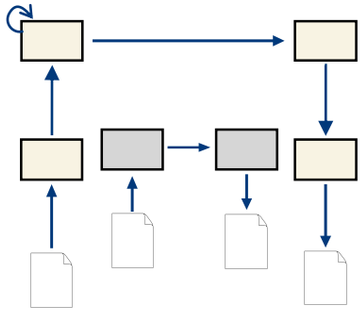
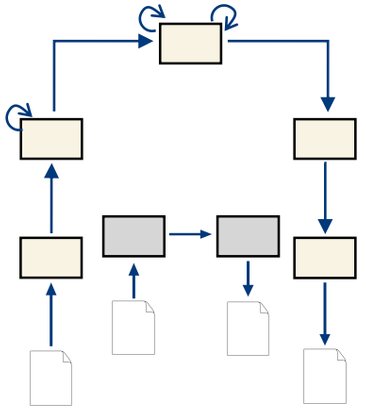
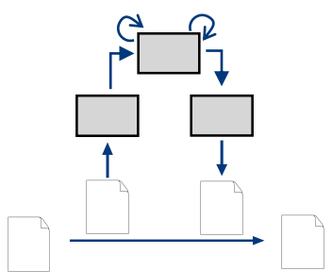
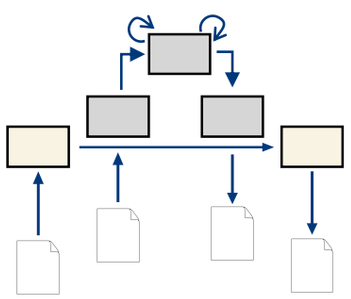
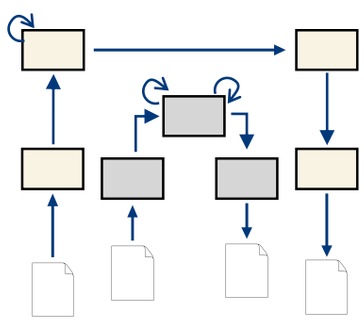
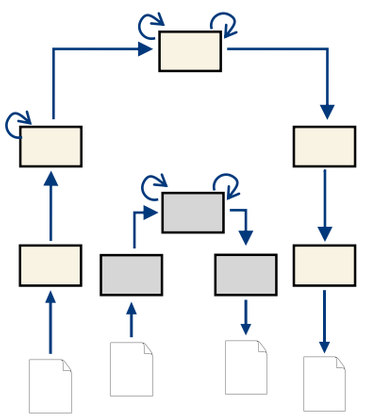




Co-Migration Patterns

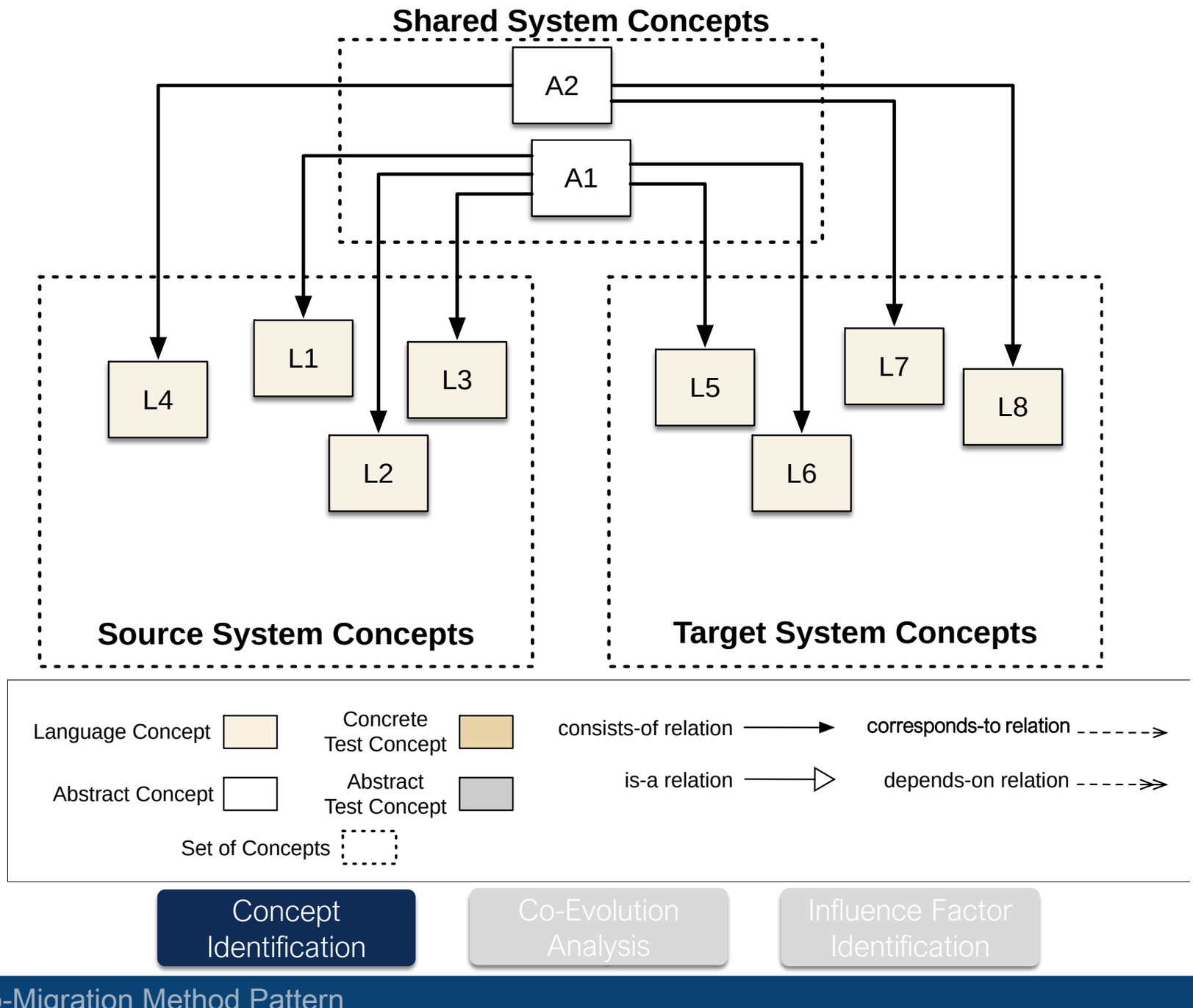
Software Innovation Campus Paderborn

Test Method Patterns

System Method Patterns	Test Reimplementation	Language-based Test Transformation	Test Language-based Test Transformation	Conceptual Test Transformation
Reimplementation				
Language-based Transformation				
Conceptual Transformation				

Concept Identification (System Concepts)

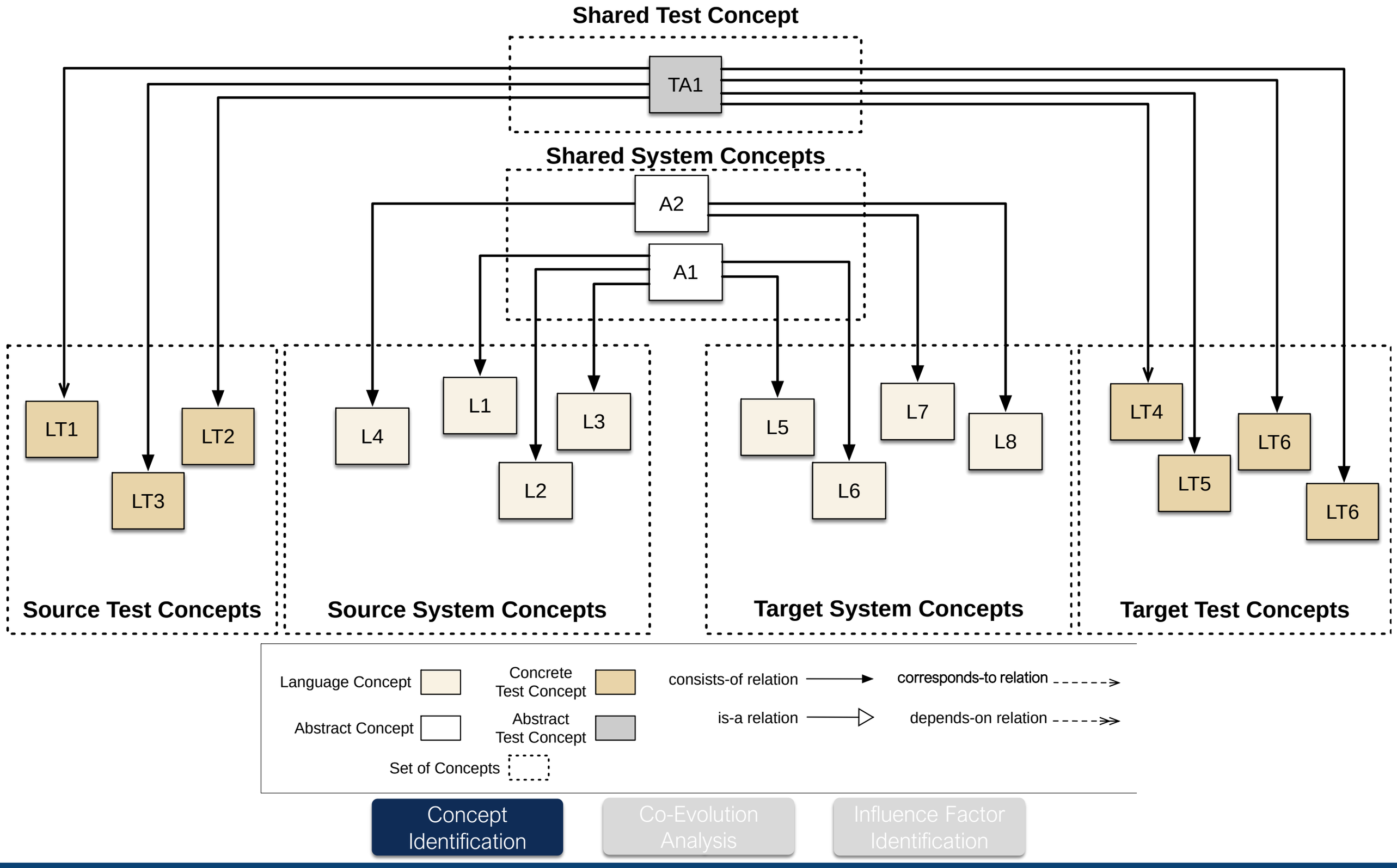
Software Innovation Campus Paderborn



Concept Identification | Co-Evolution Analysis | Influence Factor Identification

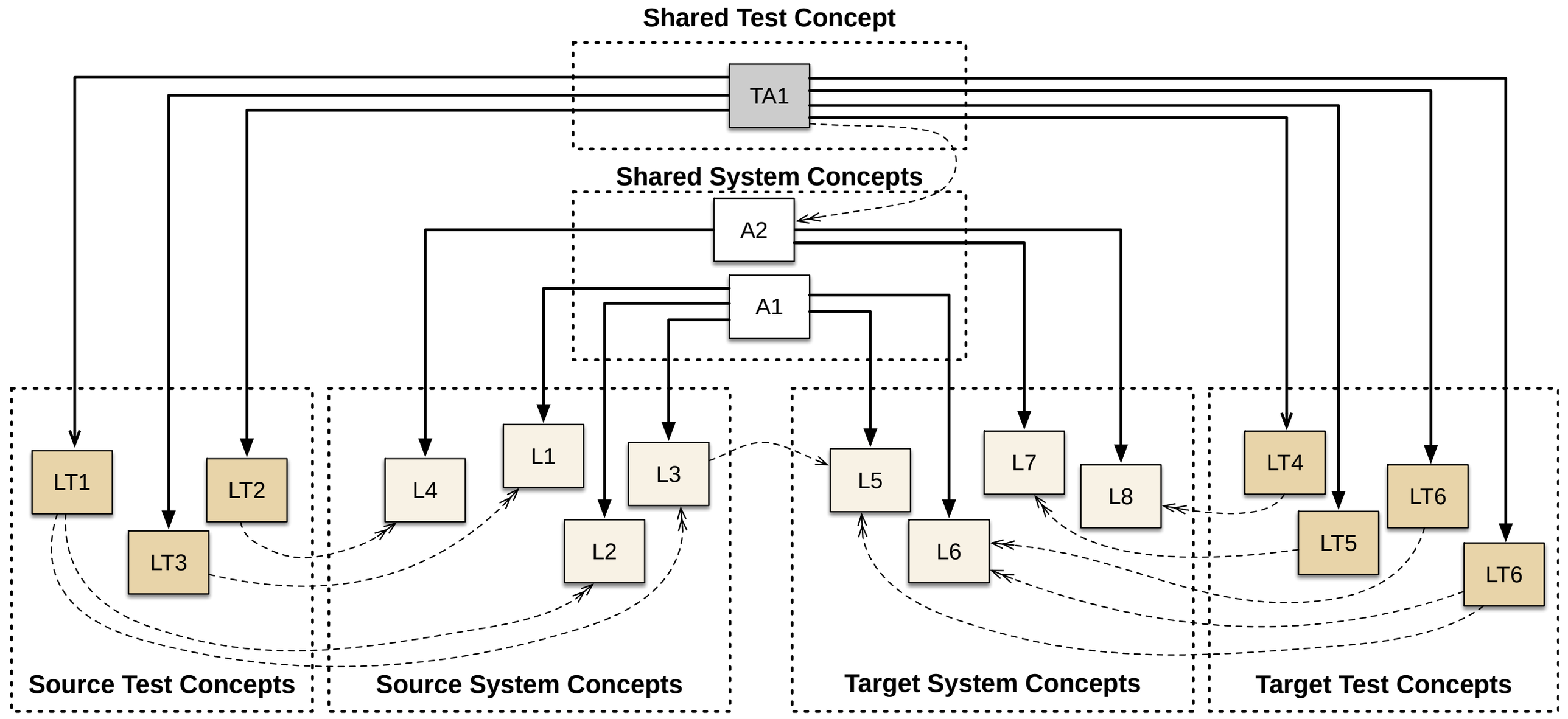
Concept Identification (Test Concepts)

Software Innovation Campus Paderborn



Concept Identification | Co-Evolution Analysis | Influence Factor Identification

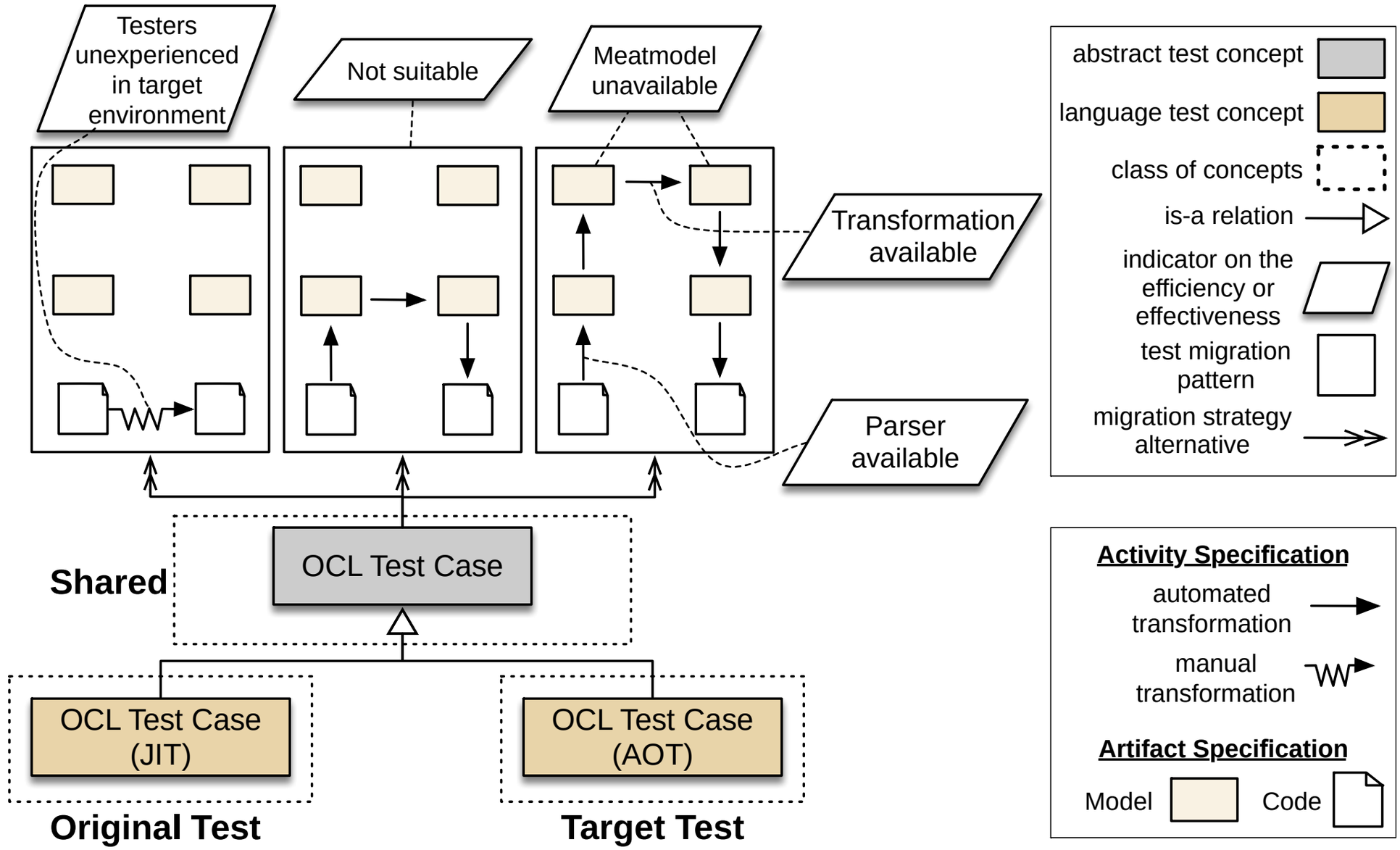
Co-Evolution Analysis



Concept Identification | **Co-Evolution Analysis** | Influence Factor Identification

Influence Factor Identification

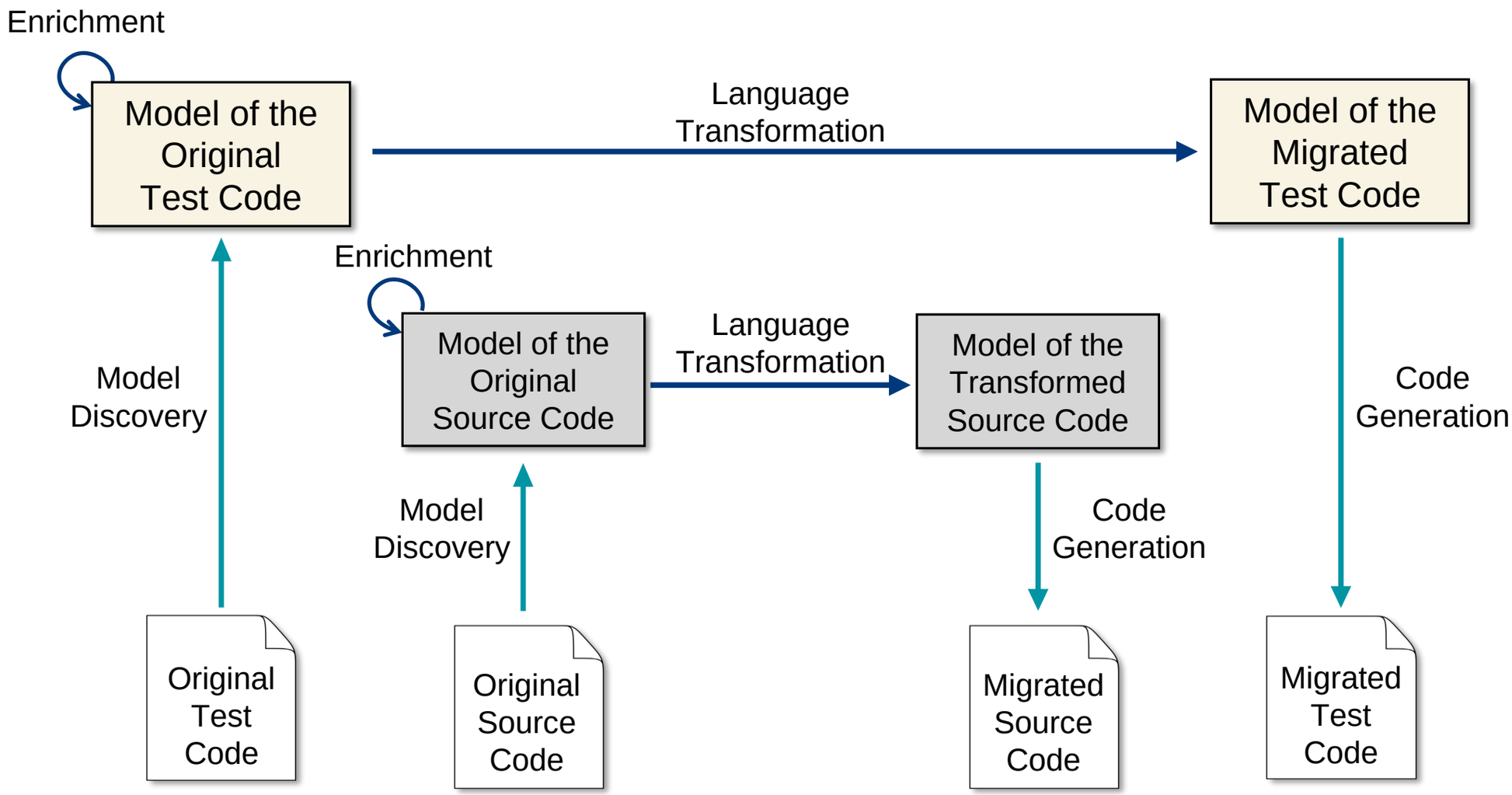
Software Innovation Campus Paderborn

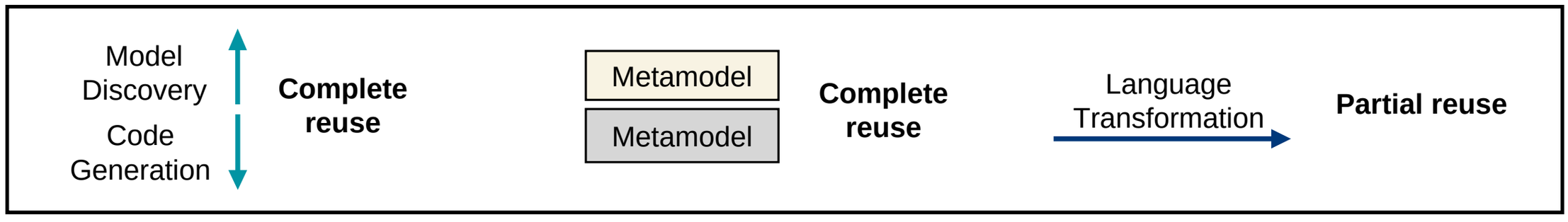


Concept Identification | Co-Evolution Analysis | Influence Factor Identification

Co-Migration Pattern – An Example

Software Innovation Campus Paderborn





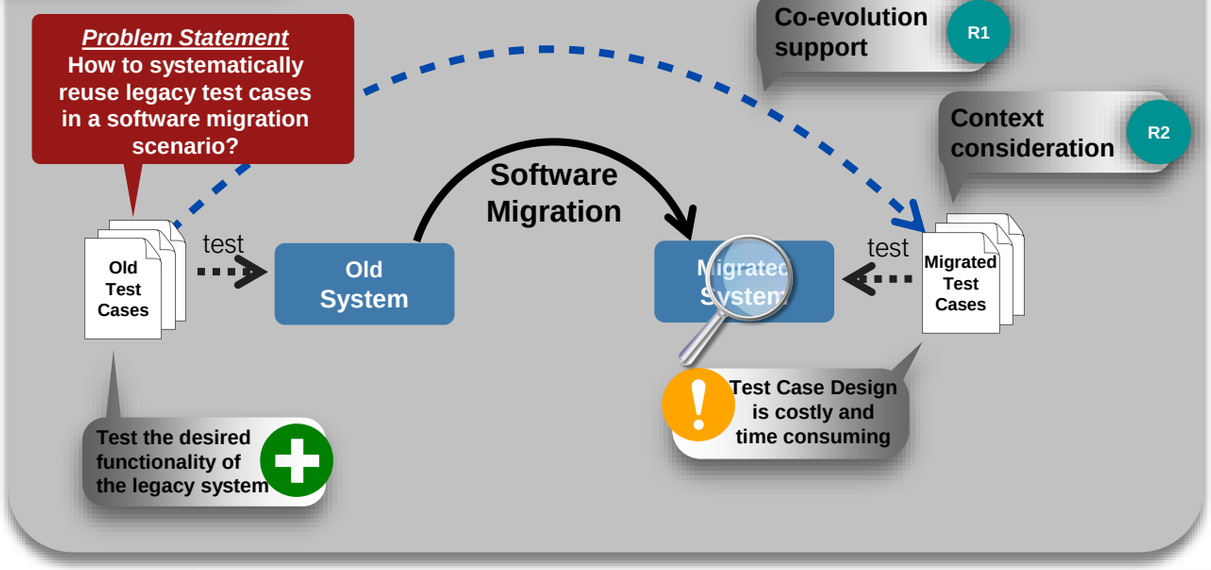
Agenda

1. **Reuse of Test Cases in Migration Projects**
2. **Test Case Co-Migration Method Patterns**
3. **Overview and Conclusion**

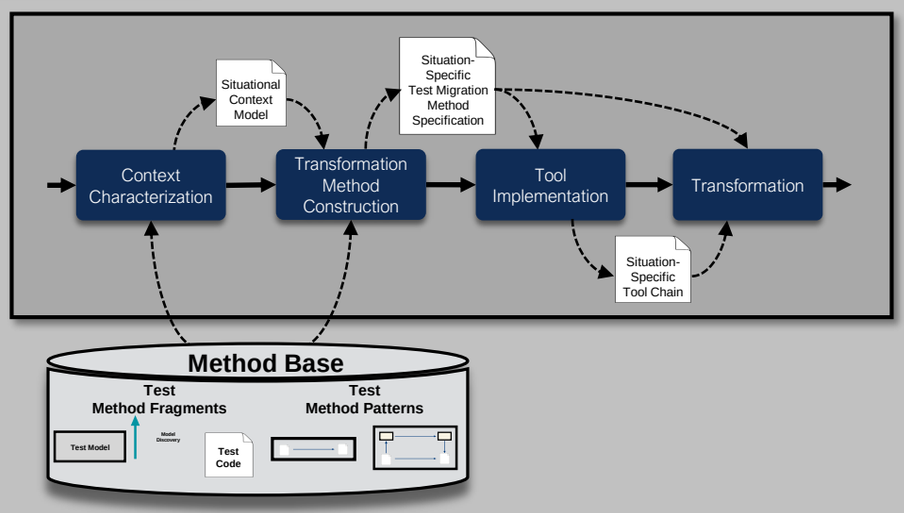
Overview and Conclusion

Software Innovation Campus Paderborn

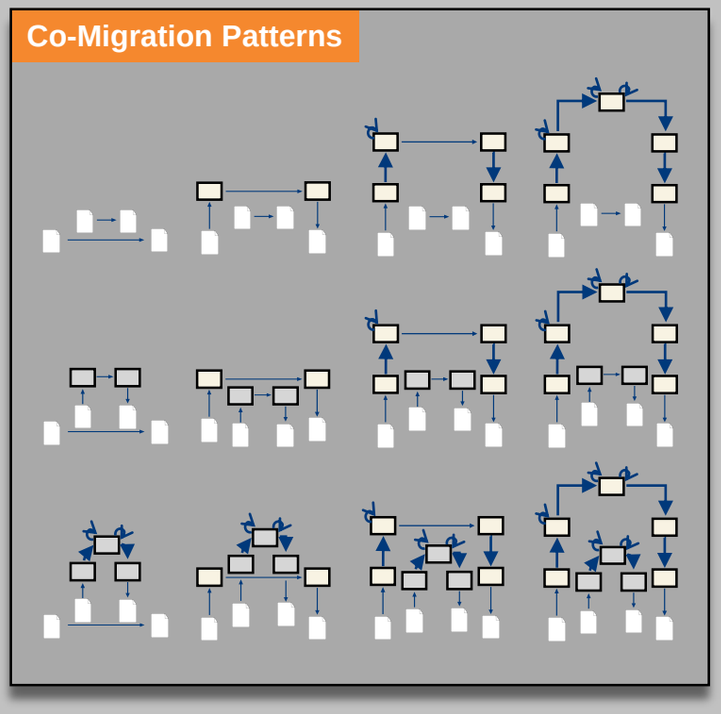
Problem Domain



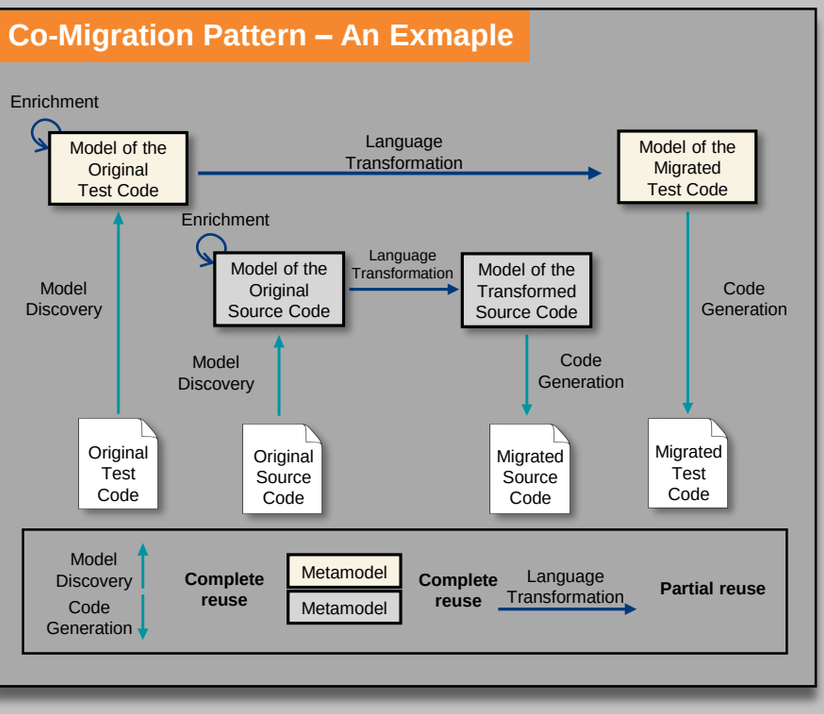
Sketched our approach



Co-Migration Patterns



Co-Migration Pattern – An Exmaple



Comprehensive discussion on the practical application

Vielen Dank für Ihre Aufmerksamkeit

SICP – Software Innovation Campus Paderborn
Universität Paderborn
Fürstenallee 11
33102 Paderborn

Ivan Jovanovikj
Tel.: (05251) 60-6841
ivan.jovanovikj@sicp.uni-paderborn.de

https://www.sicp.de/