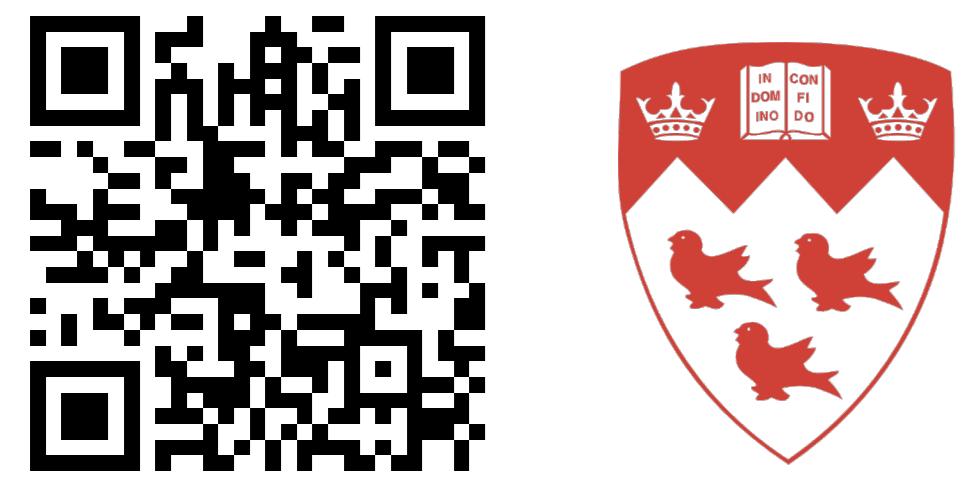


Pushing the Boundaries of Planned Reuse



McGill

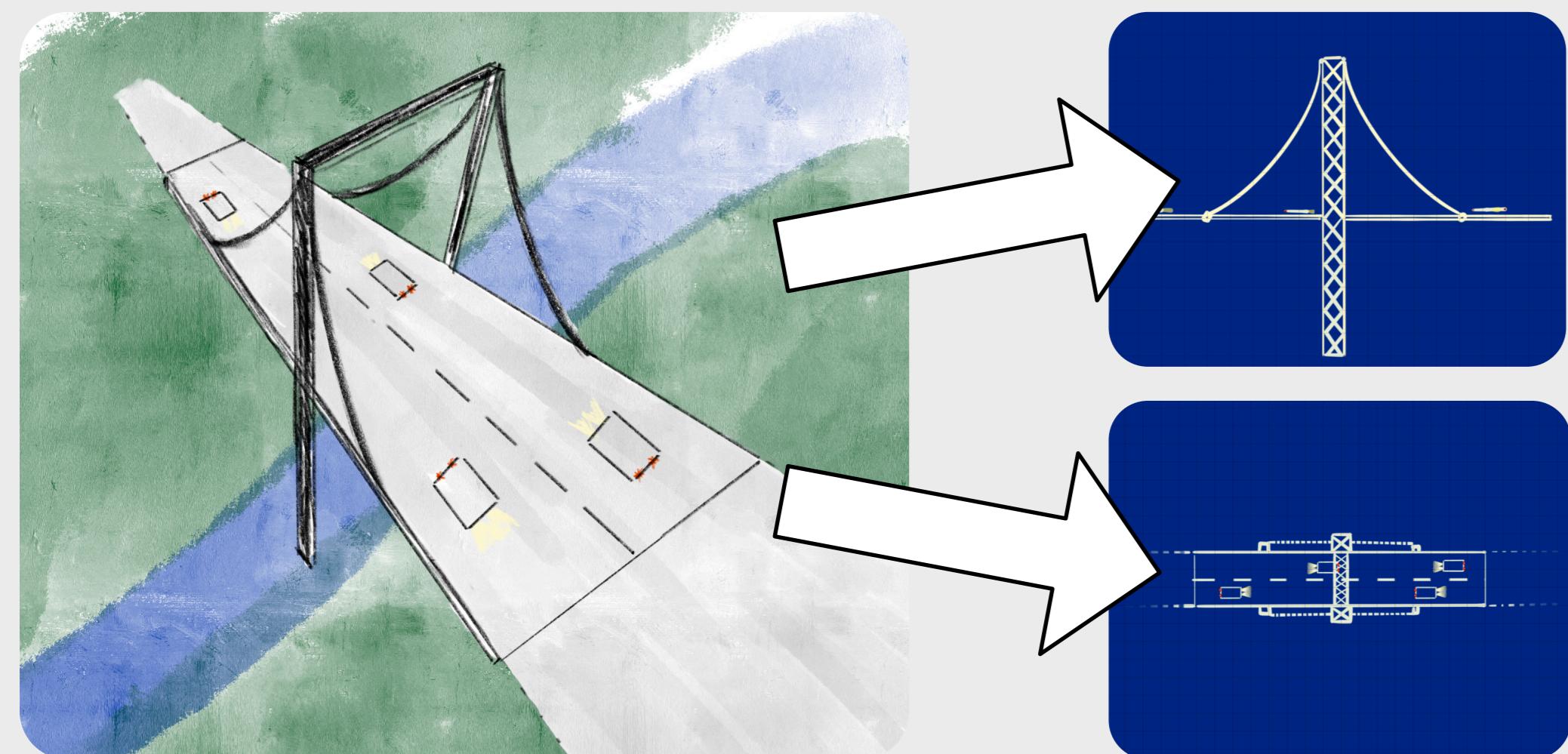
Maximilian Schiedermeier
McGill University, SCORE/DISL

SCORE
LABORATORY FOR SOFTWARE
COMPOSITION AND REUSE

Distributed
Information
Systems Lab

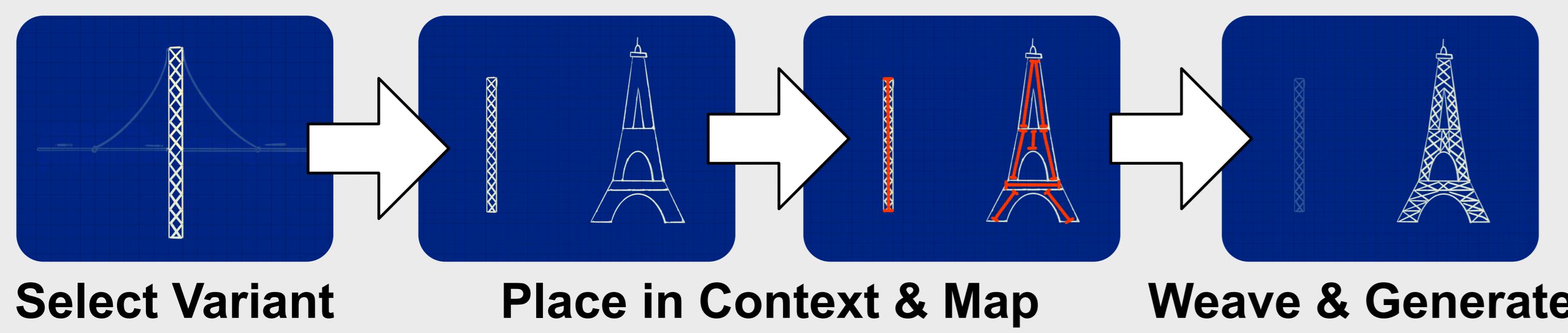
State of the Art: Reuse with GPMLs

A) Opportunistic Reuse: Extract Models



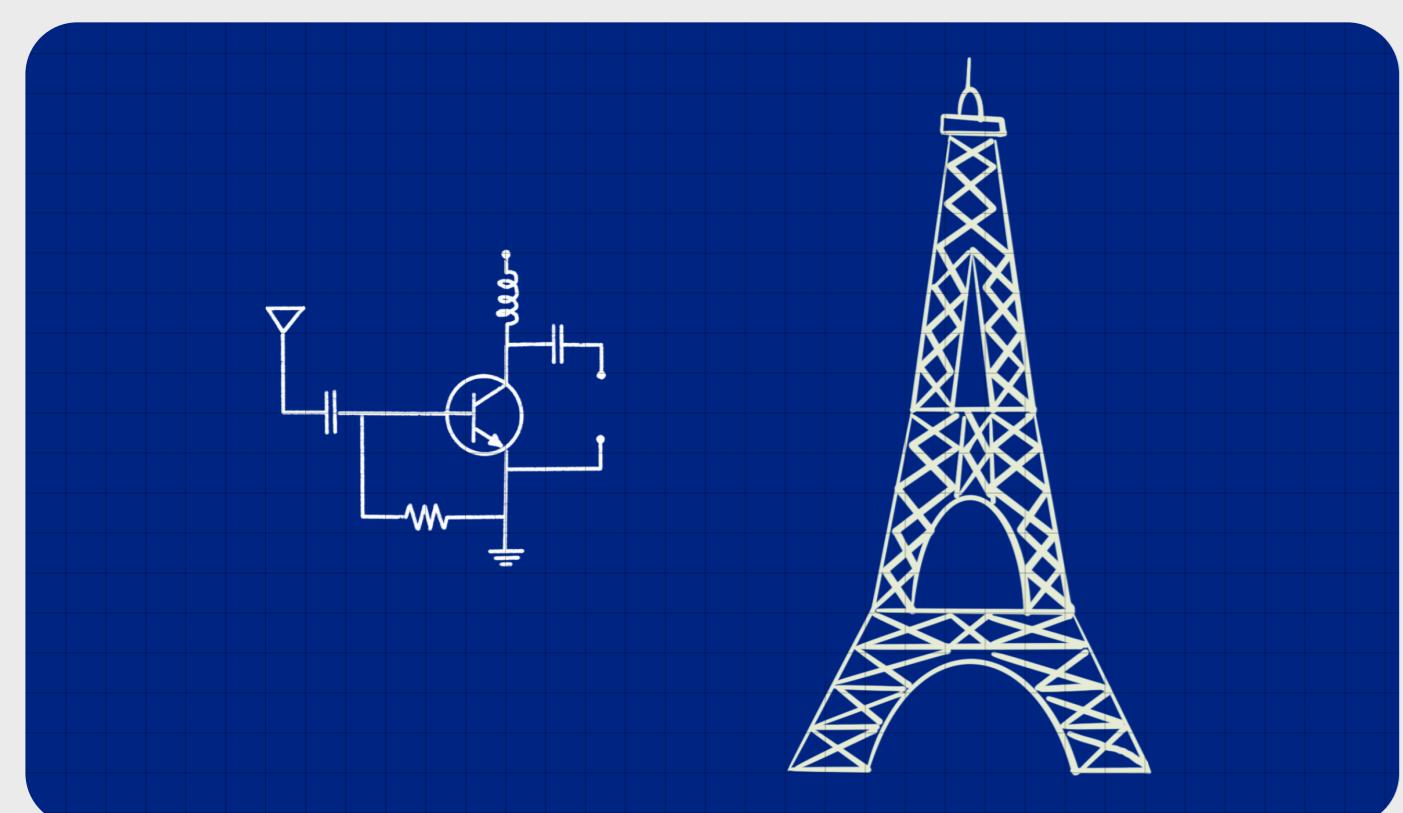
- Analyze proven structures in existing systems
- Extract as models
- Use different models for different views

B) Planned Reuse: Craft Partial Models for Reuse



Problem: DSMLs Break the Pipeline

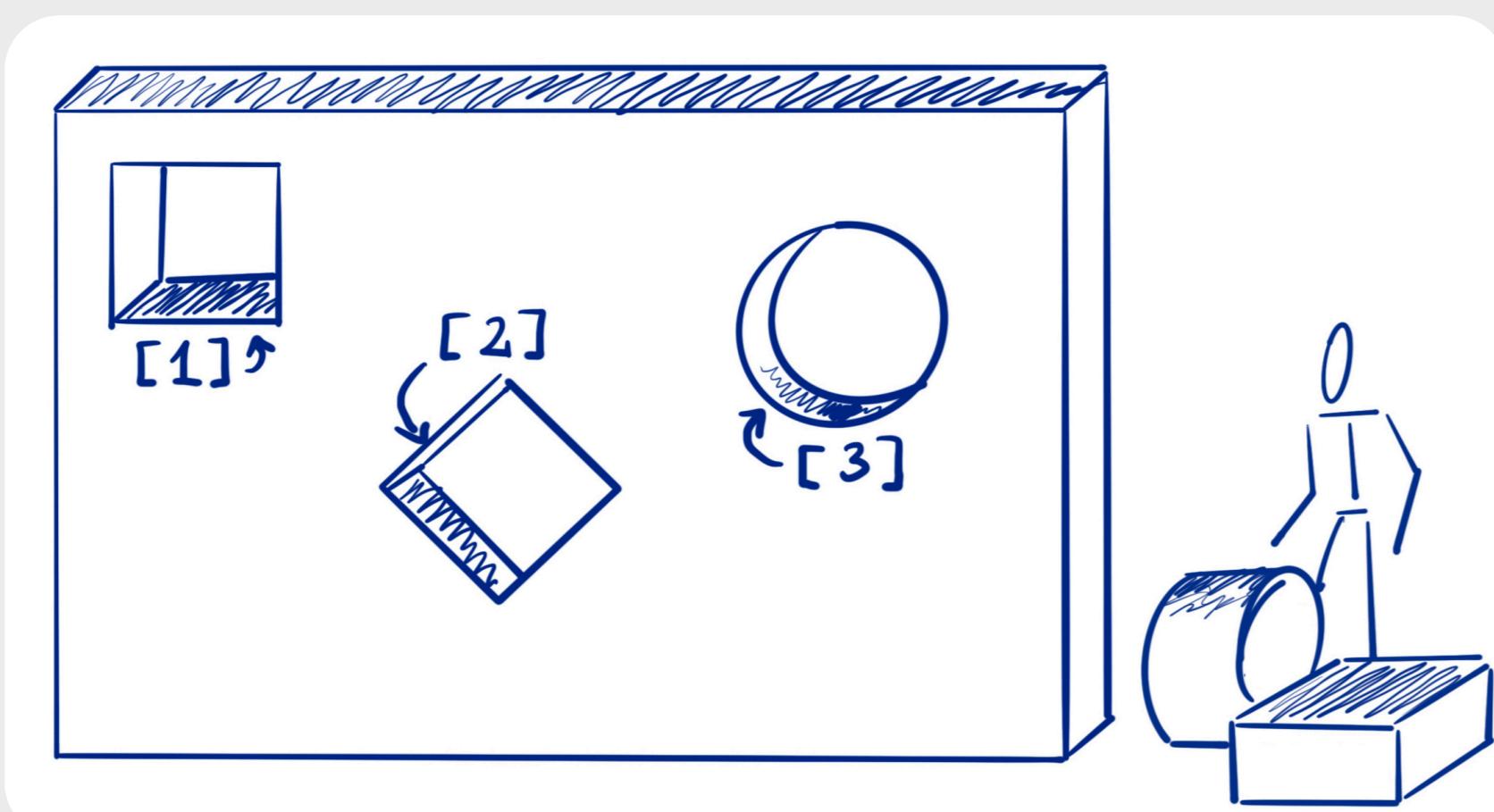
Restriction: MDE Pipeline only "works" with Compatible Models



- The matter of reuse is often hard to grasp with pure GPMLs
- DSMLs could bridge the semantic gap
- But adding a custom language breaks the MDE pipeline

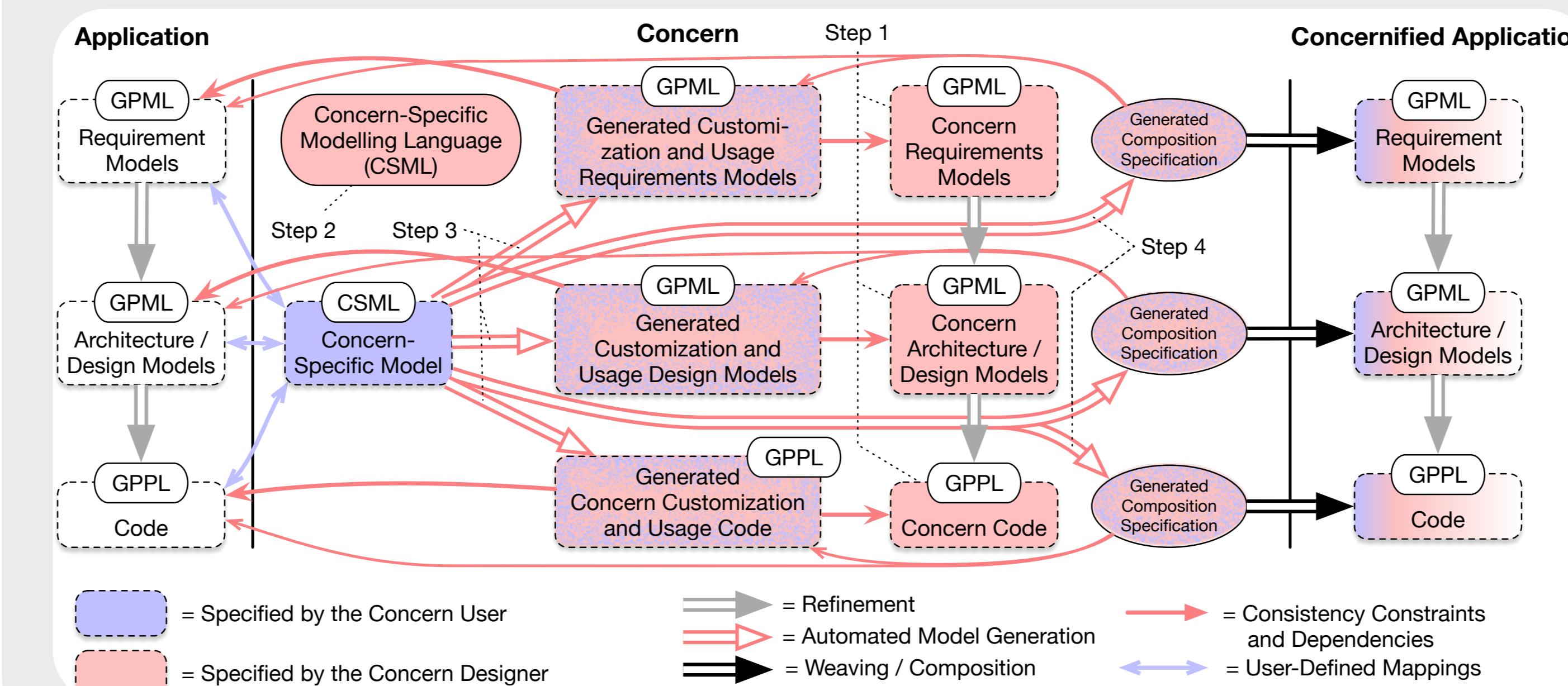
How to integrate DSMLs?

Identify and formalize a manageable **plan of action** to guide the integration of DSMLs into the pipeline...



Proposal: FIDDLR - A Methodology to fill the Blanks

FIDDLR: Framework for the Integration of Domain-Specific Modelling Languages with CORE *



*Concern-Oriented REuse (CORE) serves as a reference implementation for the MDE reuse pipeline.

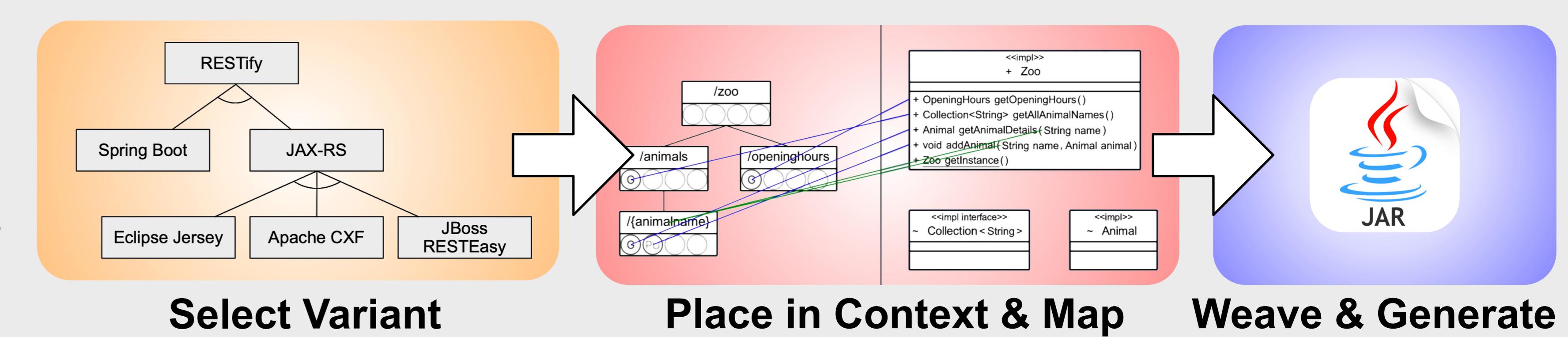
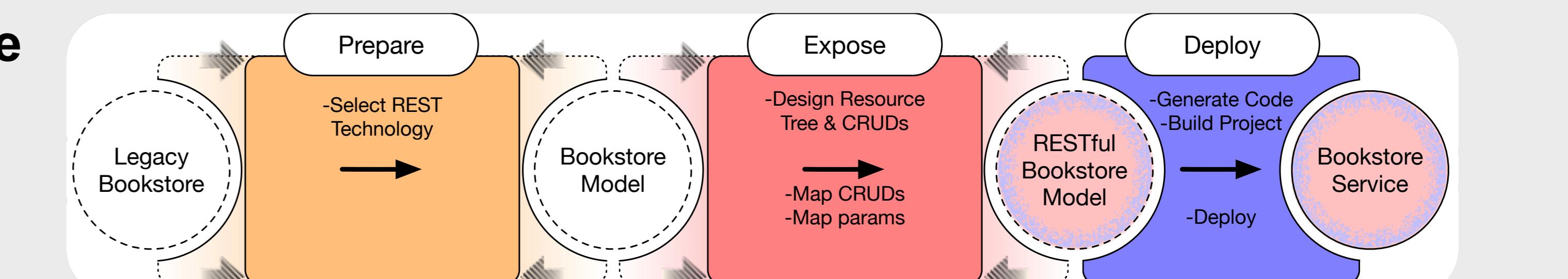
FIDDLR extends CORE by:

- A clear plan of action, separate design of:
 - A tailored DSML
 - DSML-to-GPML Transformations
 - Optional Realization Models
- Generic tooling: Generators, UI-Elements, etc...

Proof of Concept: DSML-powered Reuse with RESTify

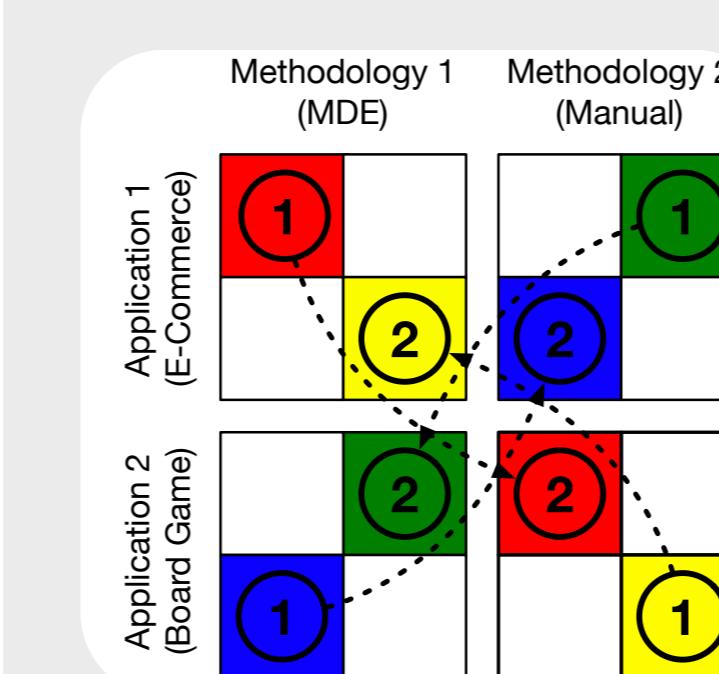
RESTify: A sample FIDDLRed MDE pipeline

- Unchains potential of DSMLs & MDE reuse
 - Assists a sample refactoring process: Converts legacy code to RESTful services
- 100% crafted with FIDDLR: comes with custom:
 - DSML to bridge semantic gap
 - DSML-to-GPML Transformations
 - Tailored Realization Models



Empiric Validation: Controlled Experiment

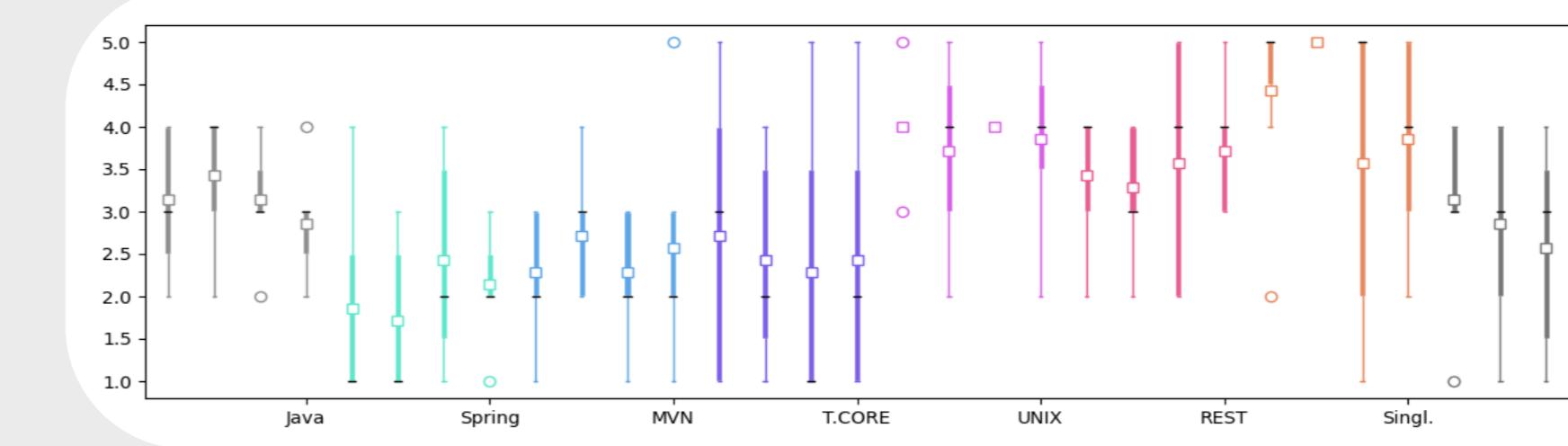
- 28 software engineers, highly diverse profiles.



4 Groups:

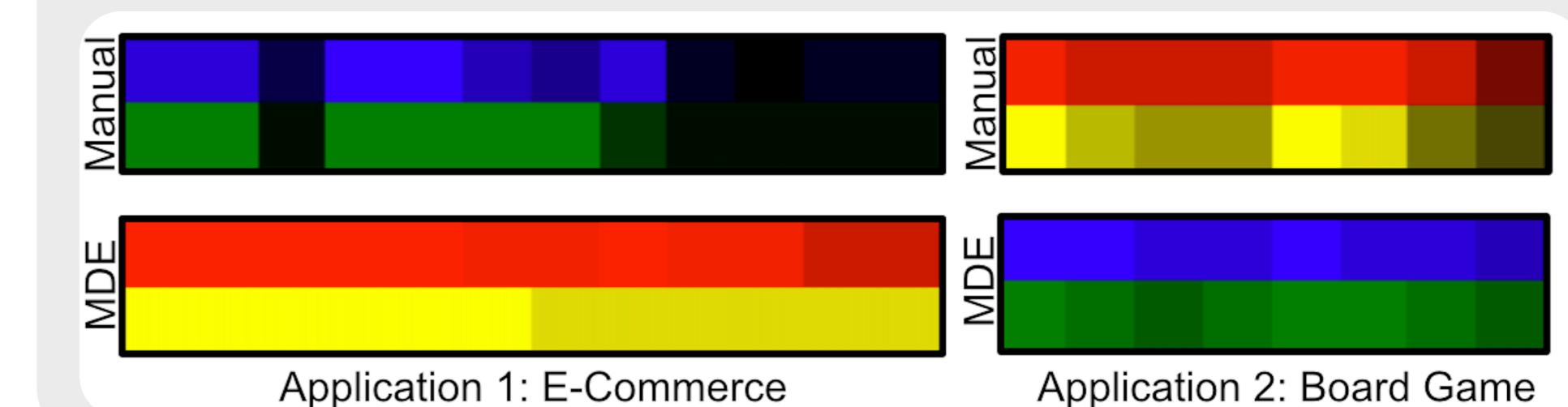
- Equal size, balanced skill vectors
 - 2 refactoring tasks per participant: Manual+MDE
 - Code + Models collected, screen recorded

A total of ~73h video material, 560 tests run!



Findings:

- FIDDLR is a viable methodology
- RESTify outperforms manual refactoring
 - Fewer test-fails than in manual tasks
 - Faster refactoring times



Below results are preliminary, we are in the process of carefully analyzing every individual submission.
43h of videos analyzed so far.
Can present preliminary CSV data to support claims.