MoDELS 2007

Model-Driven Engineering for Software Migration in a Large Industrial Context

Franck Fleurey,
Erwan Breton,
Benoit Baudry,
Alain Nicolas,
Jean-Marc Jézéquel

Context and motivations

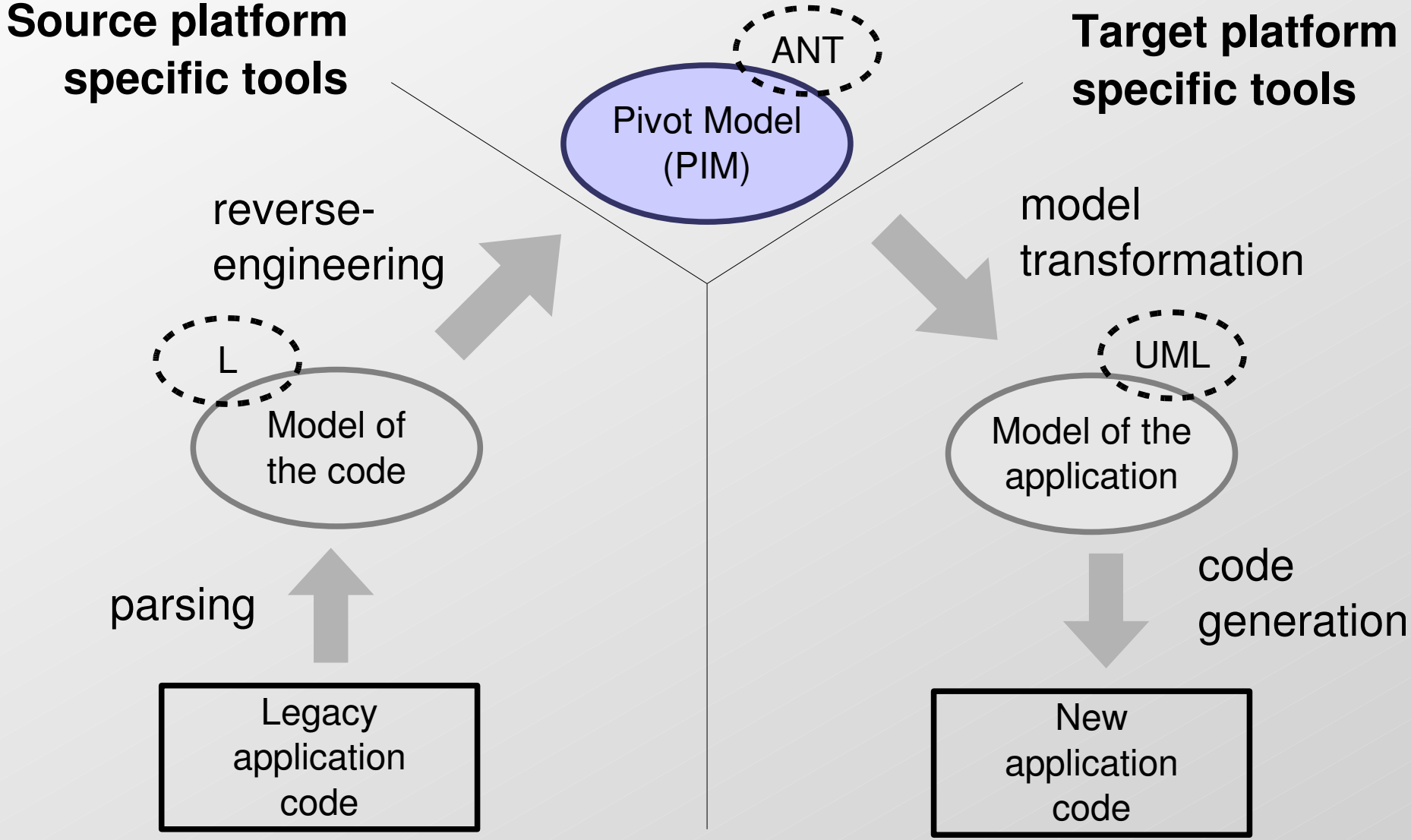
- Migration of legacy applications
  - No change in the requirements
  - Migration to new technology platforms
  - Refactoring in the design
  - e.g. banking, assurance, ...
- Two competing approaches
  - Full re-development (outsourcing)
  - Semi-automated migration

SOFT-MAINT uses MDE for automating migration

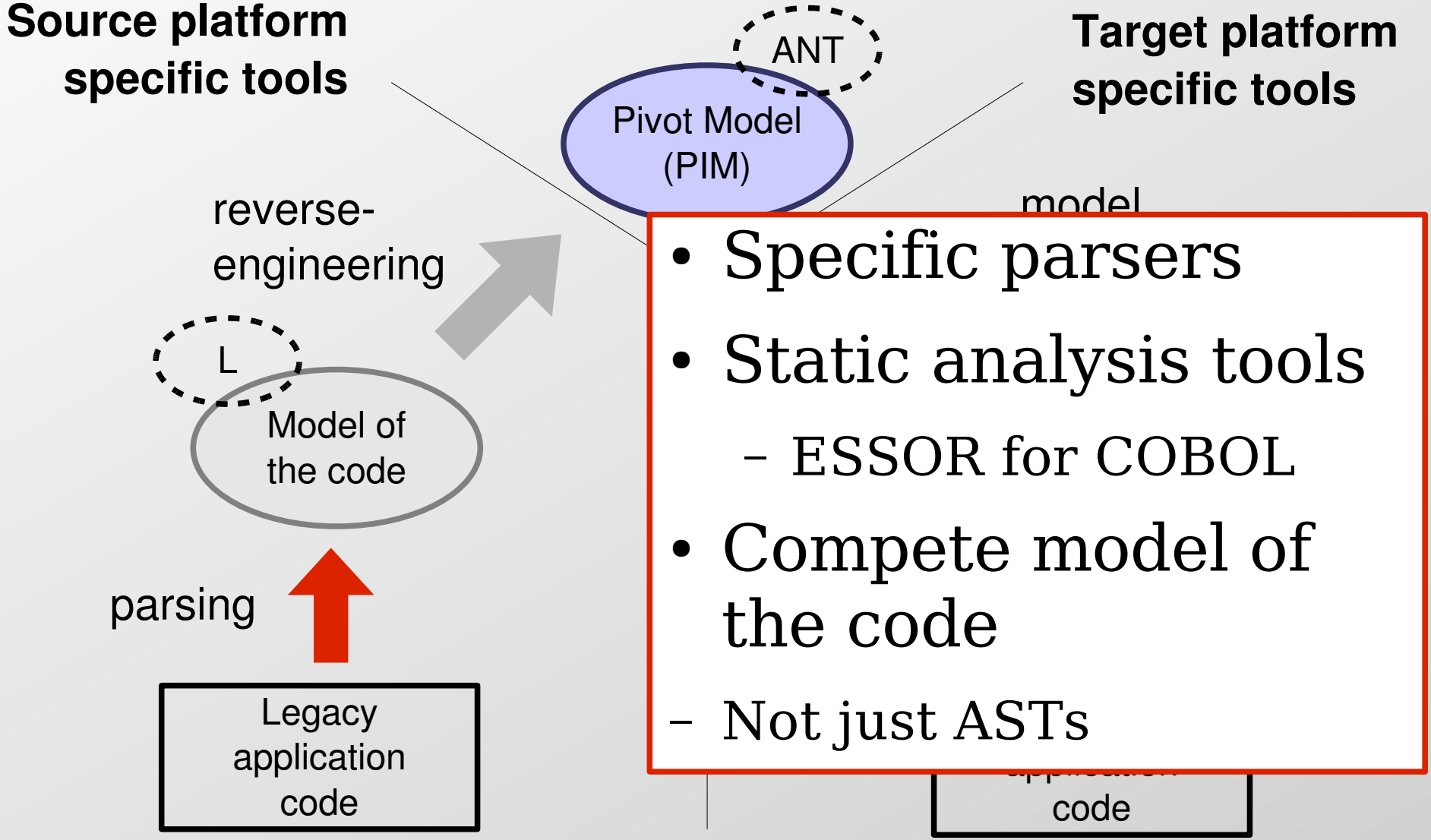
Outline

- Model-based software migration
- Migration projects phases
- Case Study
- Comparison with re-development
- Conclusion and limitations

Model-based software migration

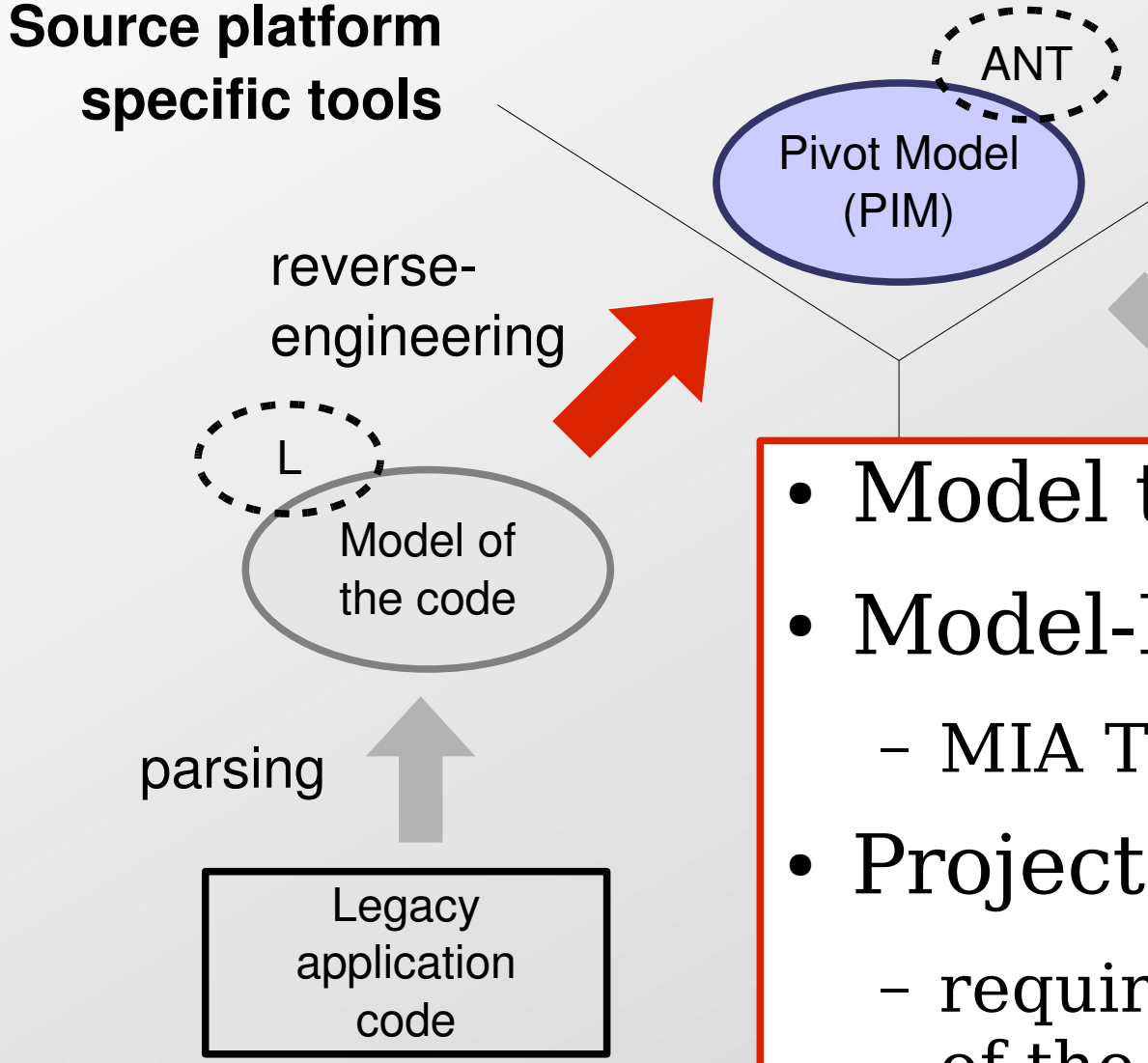


1. Parsing the legacy code



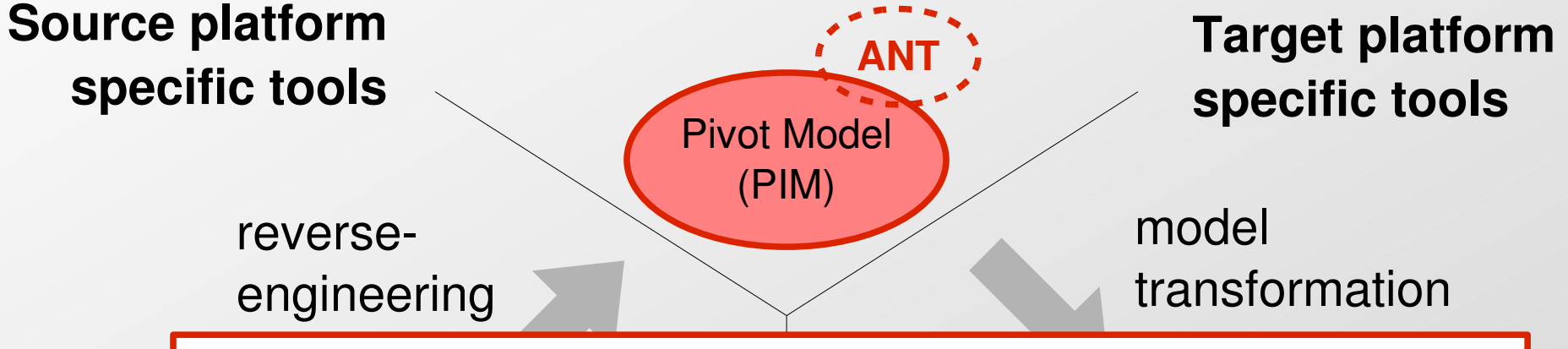
- Specific parsers
- Static analysis tools
  - ESSOR for COBOL
- Compete model of the code
- Not just ASTs

2. Reverse engineering



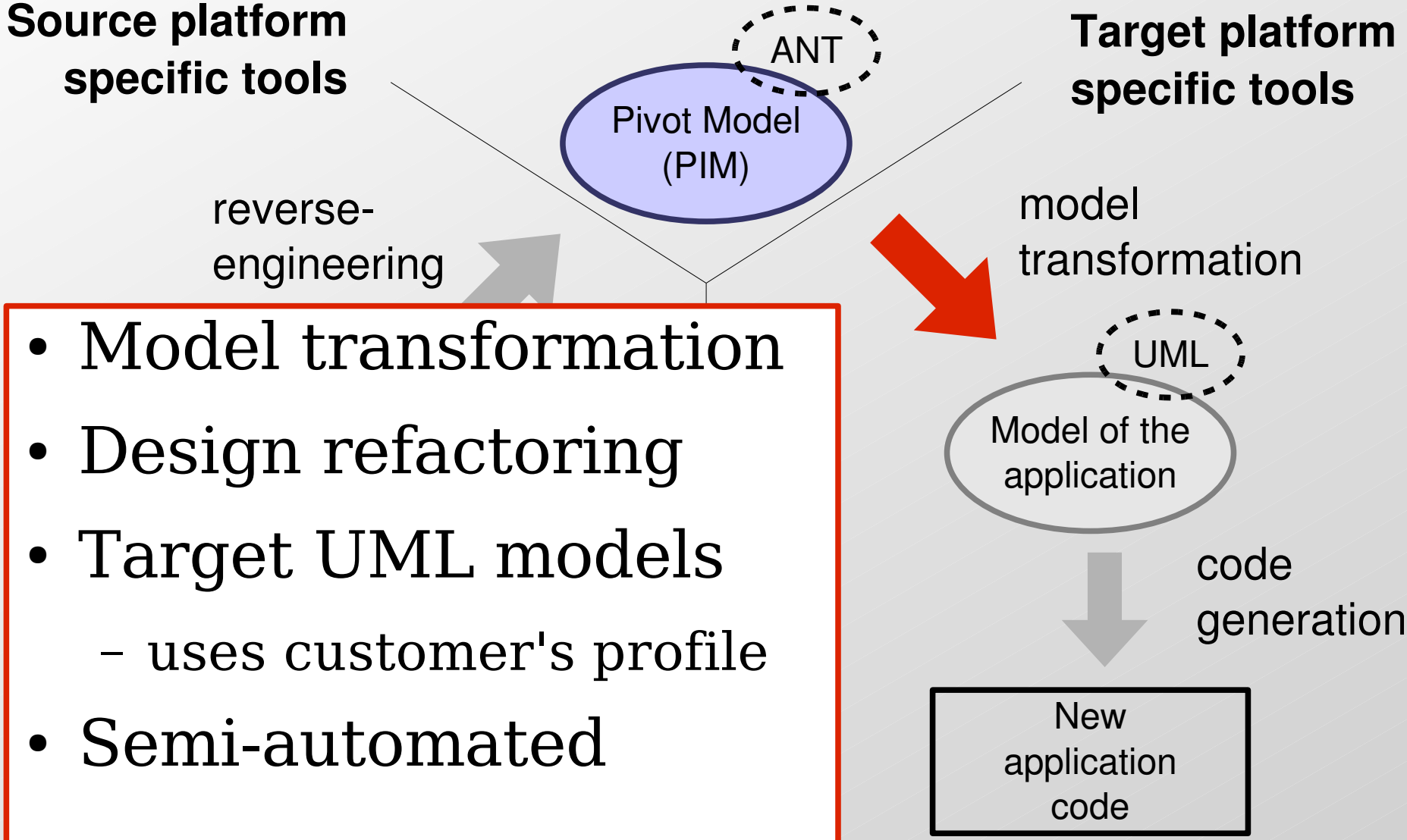
- Model transformation
- Model-In-Action tools
  - MIA Transformation
- Project specific
  - requires a deep analysis of the legacy application

The pivot model



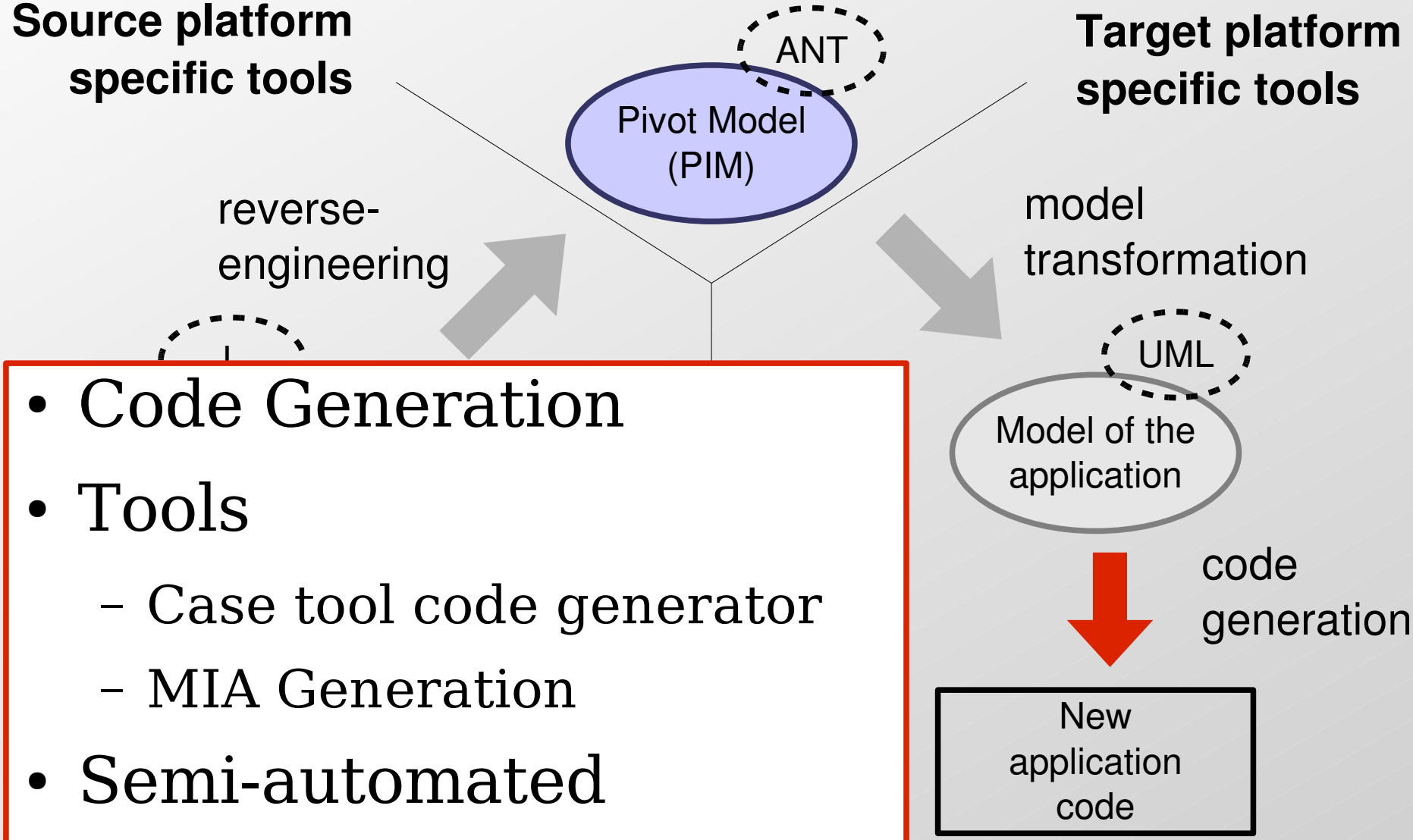
- Reusable pivot model
- Capture all the information
  - Static structures (Class diagrams)
  - Graphical User Interface
  - Application navigation (work flow)
  - Actions and algorithms

3. Model transformation



- Model transformation
- Design refactoring
- Target UML models
  - uses customer's profile
- Semi-automated

4. Code Generation



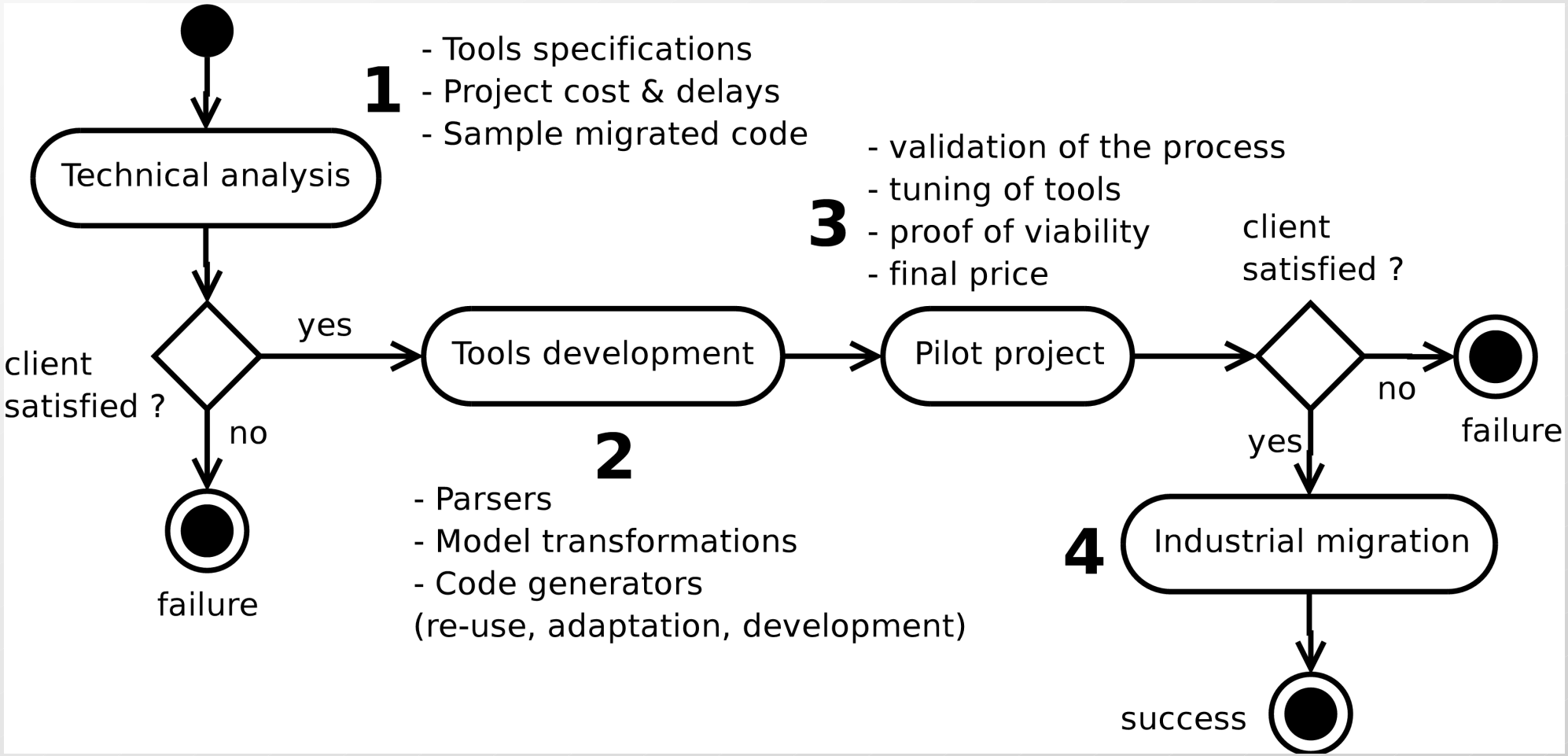
- Code Generation
- Tools
  - Case tool code generator
  - MIA Generation
- Semi-automated

Implementing the migration process

- Requires generic tools
  - legacy language specific tools
  - model transformation tools
  - code generation tools
- Requires project specific tools
  - reverse-engineering transformations
  - design transformations
  - code generators

Significant preliminary work

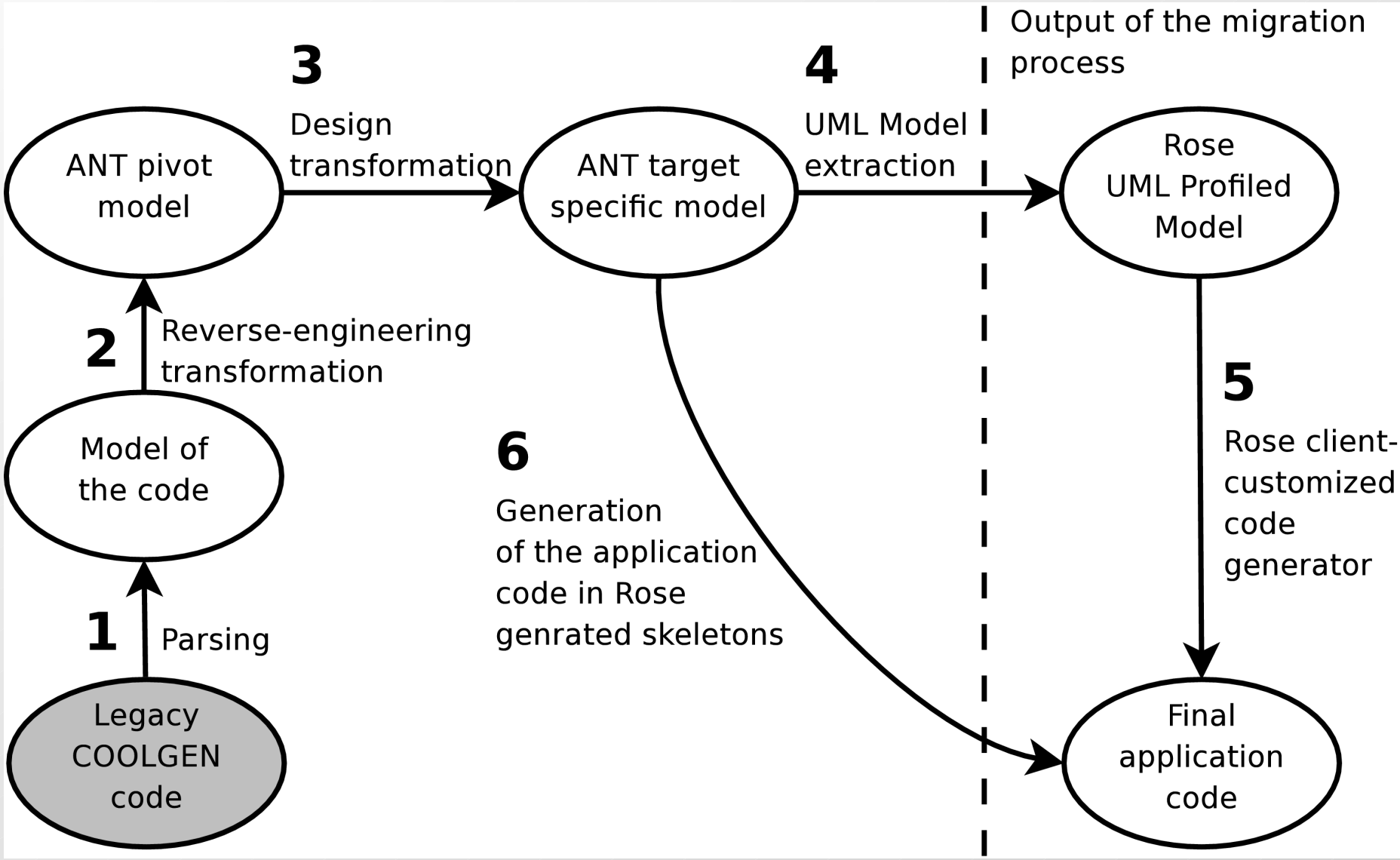
Preliminary phases of migration projects



Case study

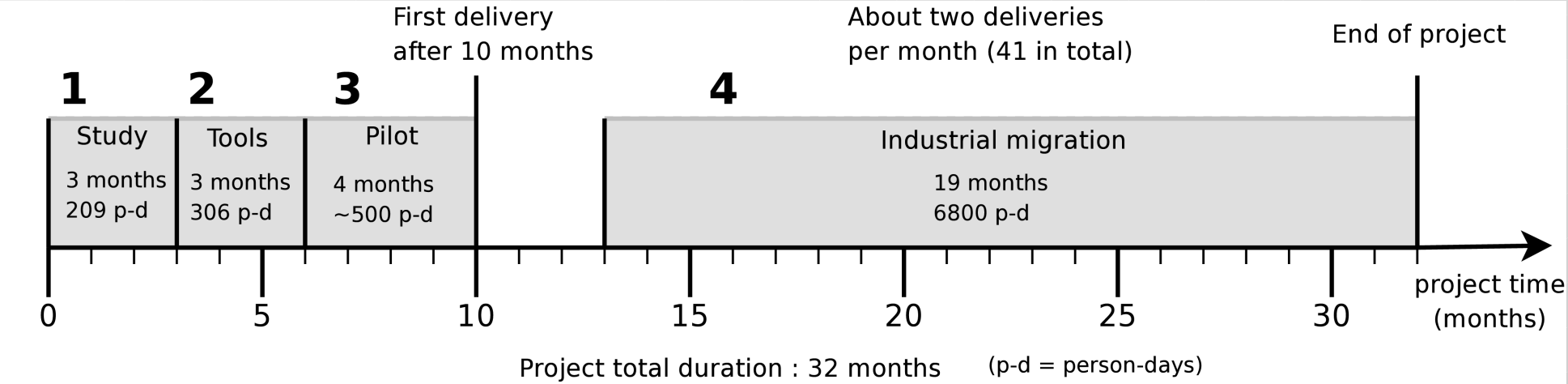
- Banking system
  - 1 million lines of COOLGEN code
  - 42 applications (800 forms, 7500 events)
  - 99 Crystal Report exports
  - 990 server services
  - 20 batch processes
- Requirements
  - Servers in COBOL
  - Applications in J2EE
  - Round-trip custom UML models

Specific migration process



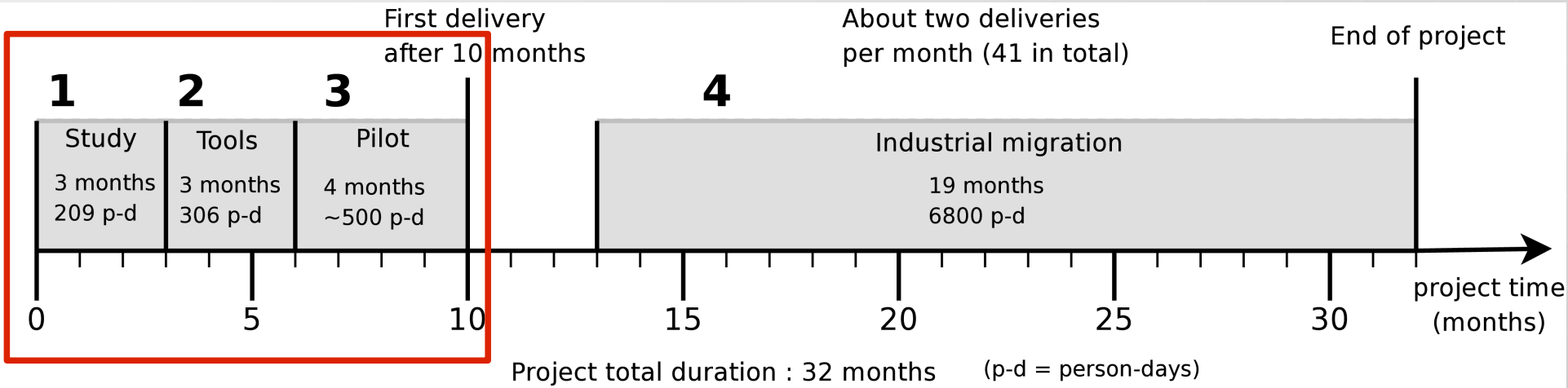
Migration project schedule

- 32 months total
- 9315 person-days
- 10 month preliminary work



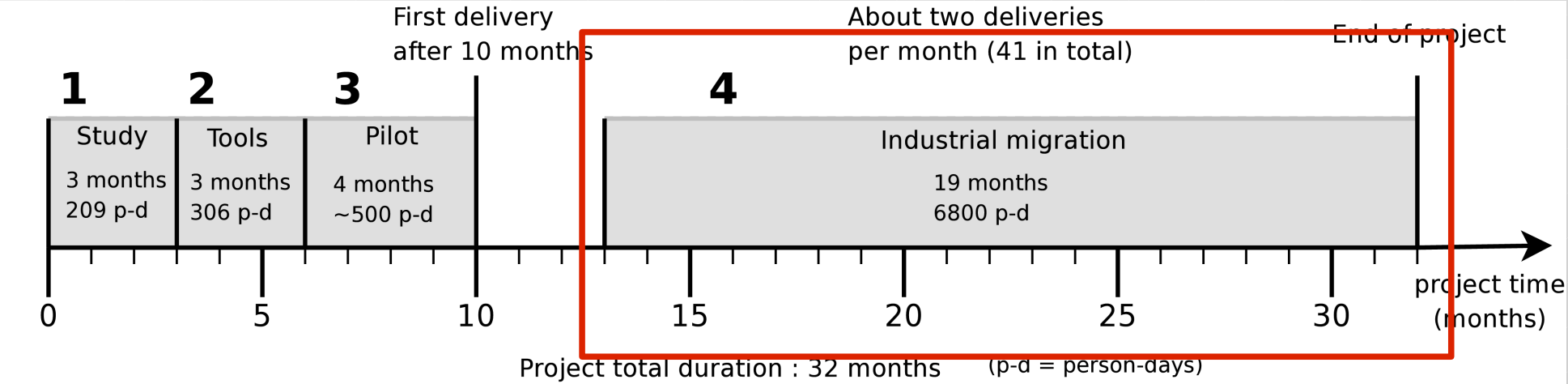
Preliminary tasks

- Small developer team (3 to 8 persons)
- 10 months
- 1000 person-days
- 12% of the project cost



Industrial migration

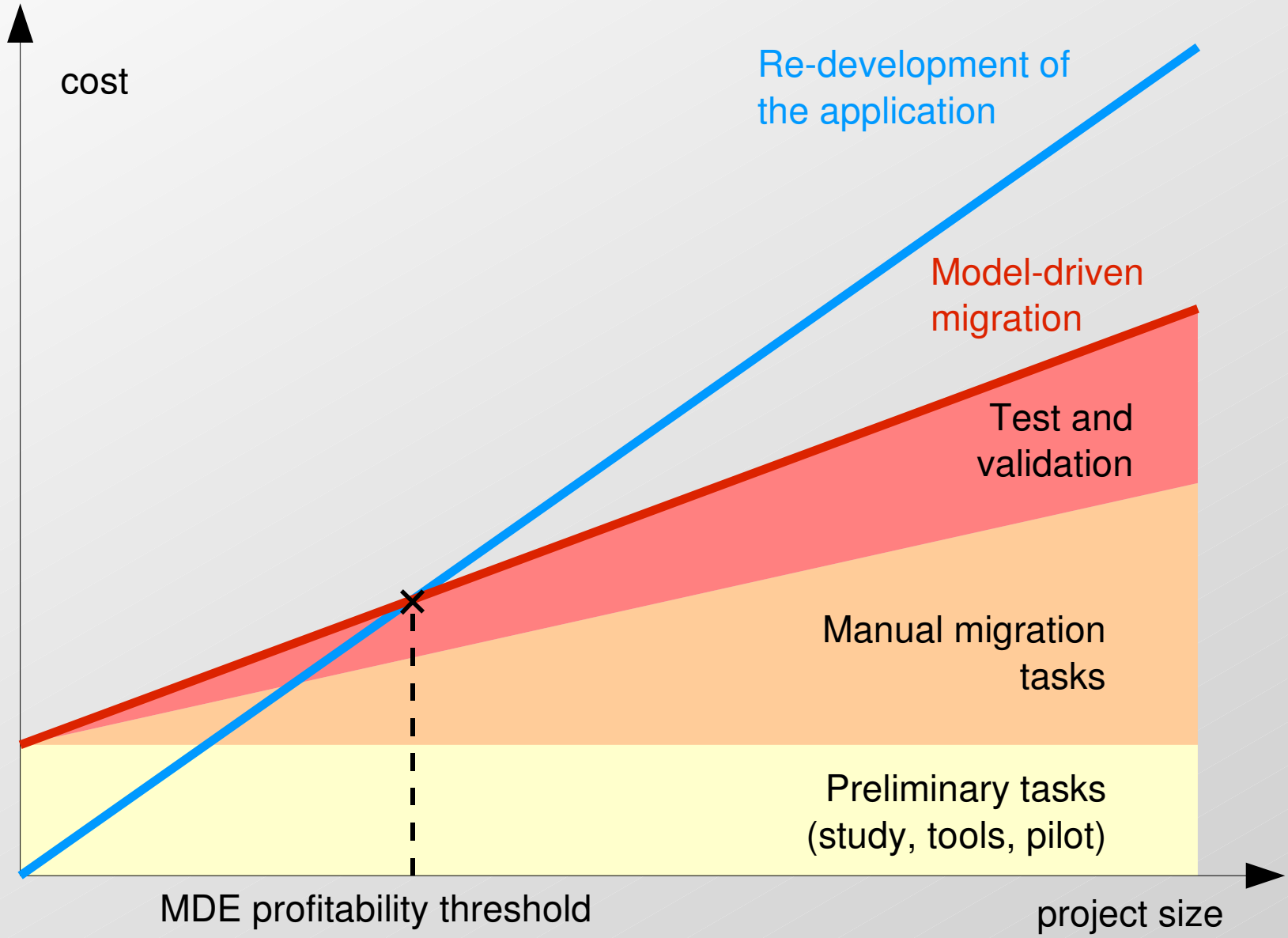
- 3 parallel teams of 15 developers
- 19 months
- 6800 person-days
- 2 deliveries per month in average



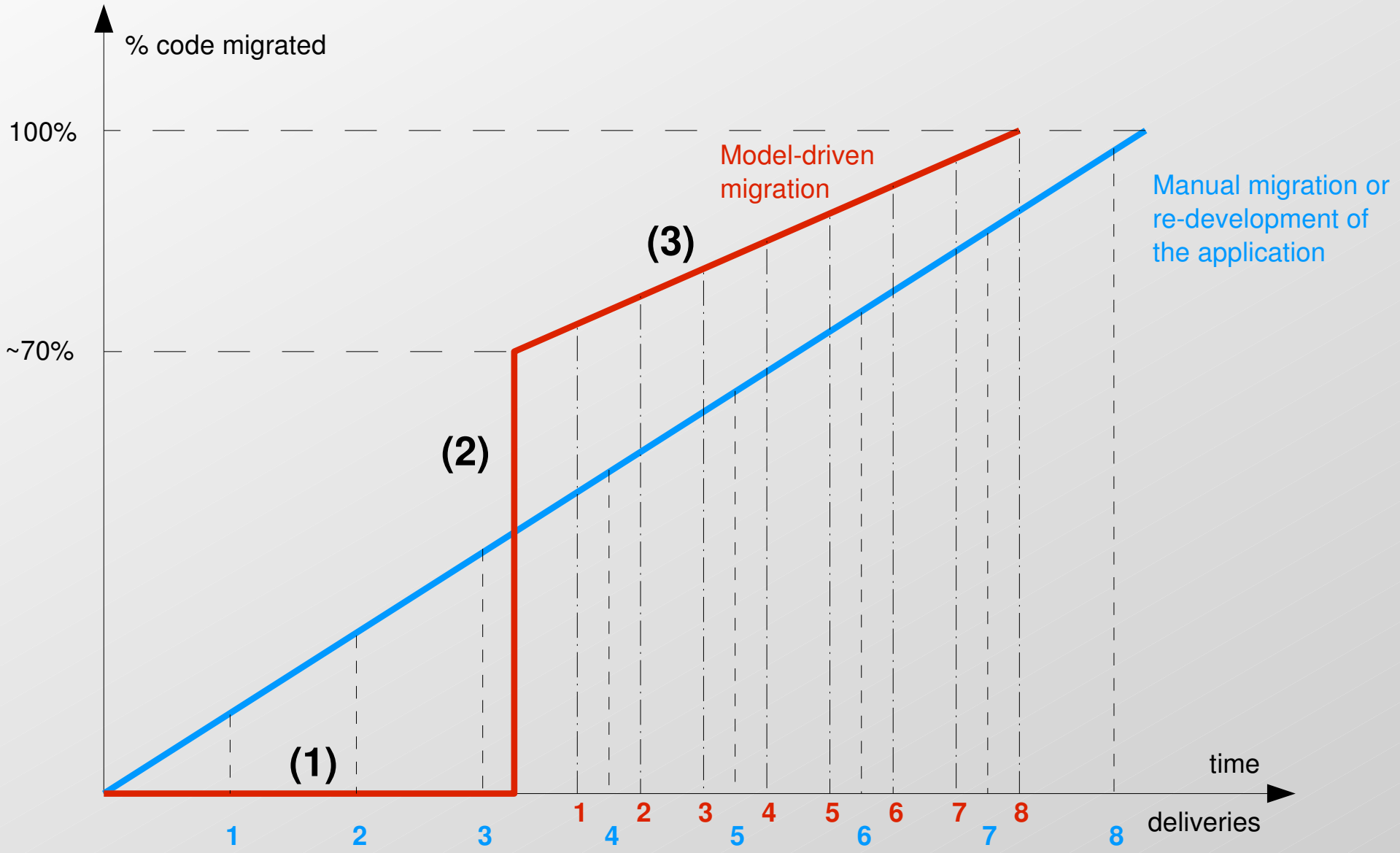
Case study results

- Cheaper than re-development
  - About half the cost
- High level of automation
  - 70% of the final code is generated
- Validation is expensive
  - about 50% of the overall cost

Discussion (1)
Profitability vs. project size



Discussion (2)
Migration time-schedule



Summary and conclusion

- Allows for automation
- Allows reuse of tools / transformation
- Profitable over re-development
- Long preliminary phase required
  - Commercial limitation
- Test represents 50% of the manual work
  - Regression testing

Questions ?