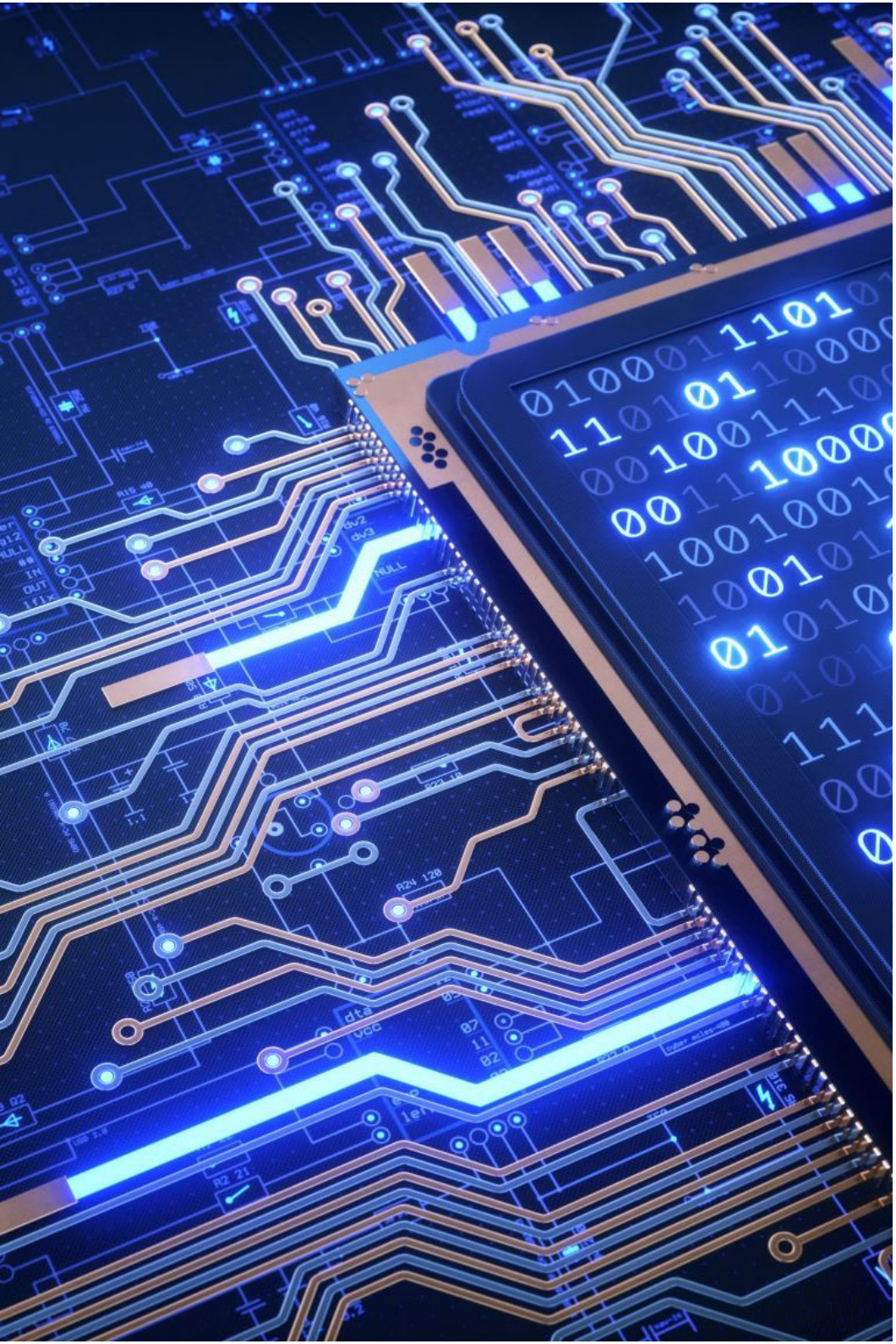
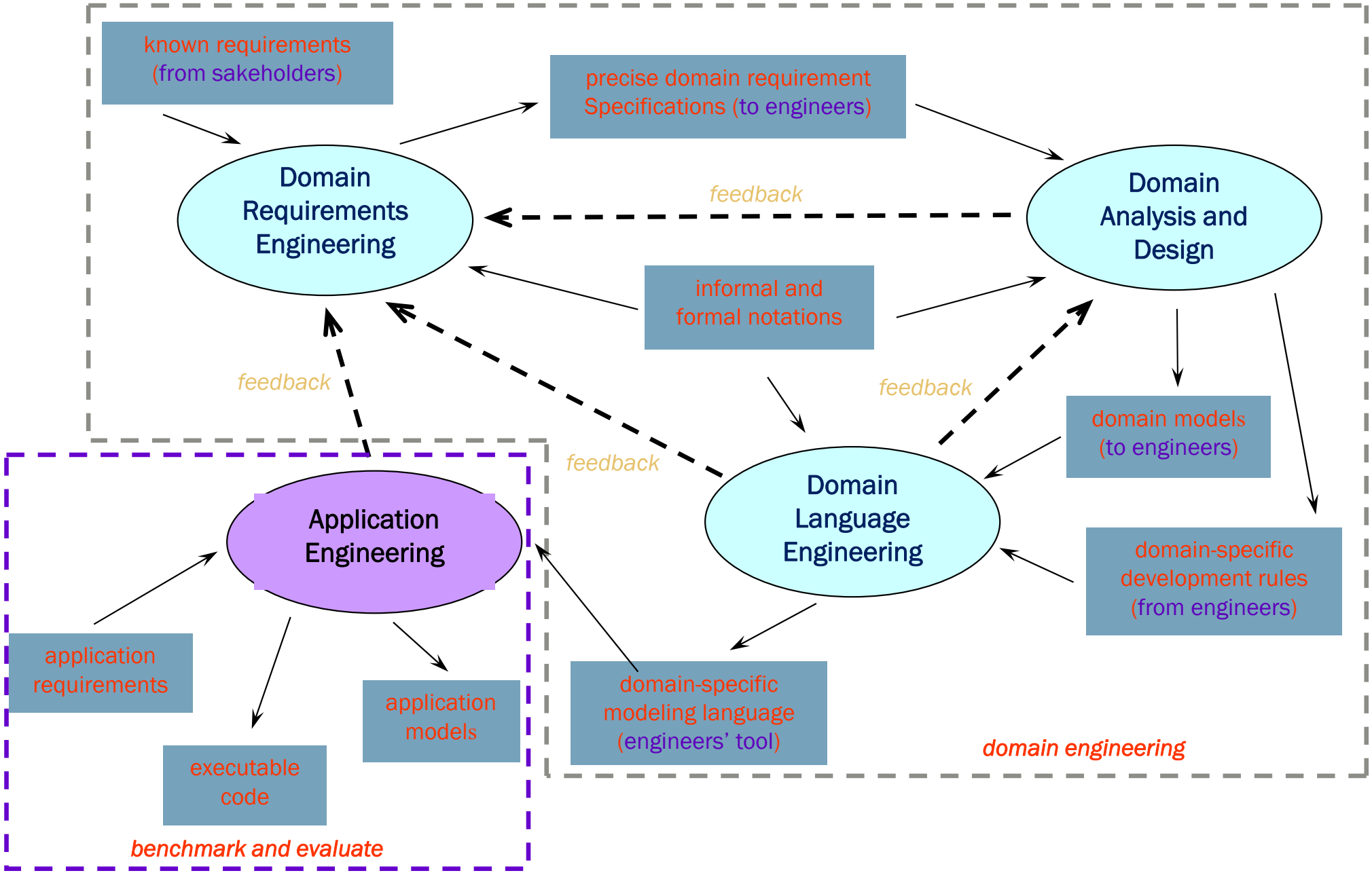


CSci363 User Interface Design

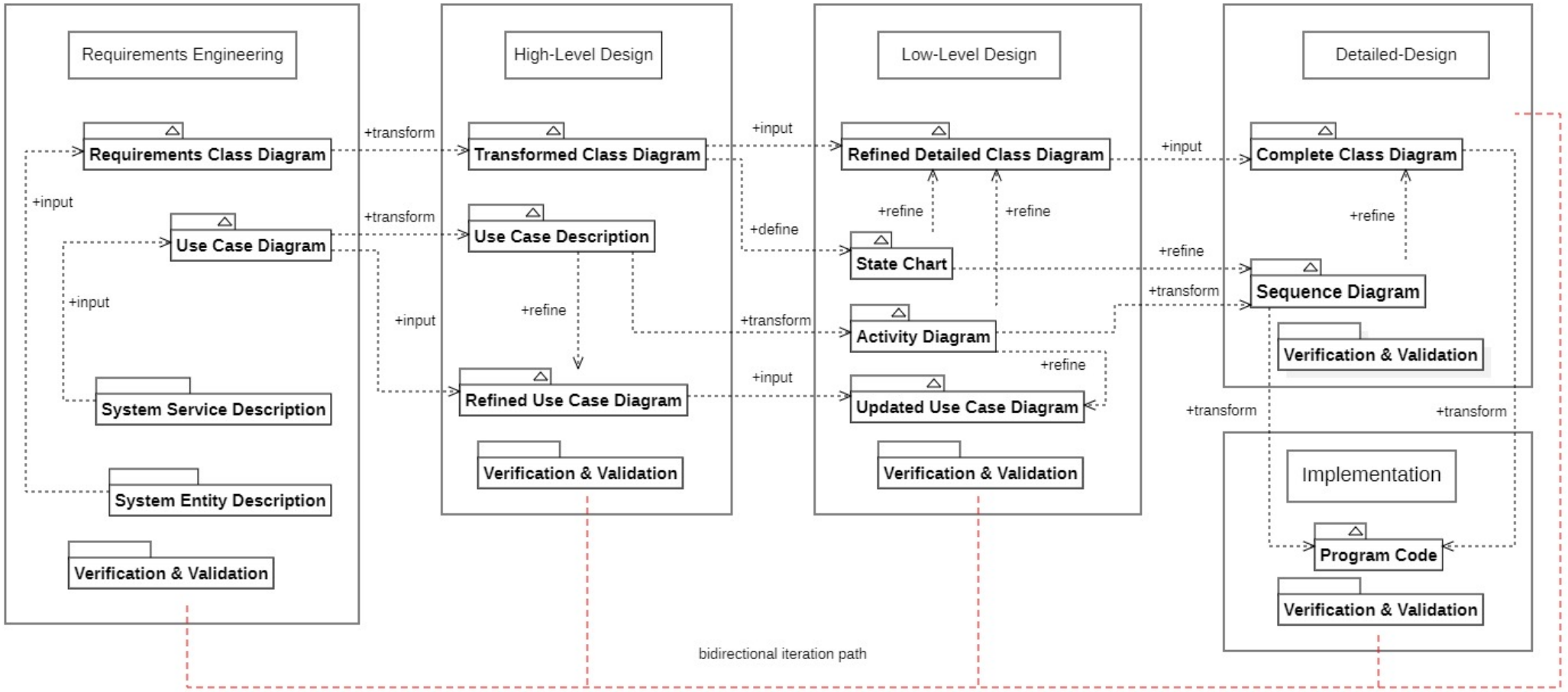
Monday, September 9, 2024

- **Today's session**:
- Software Engineering Principles

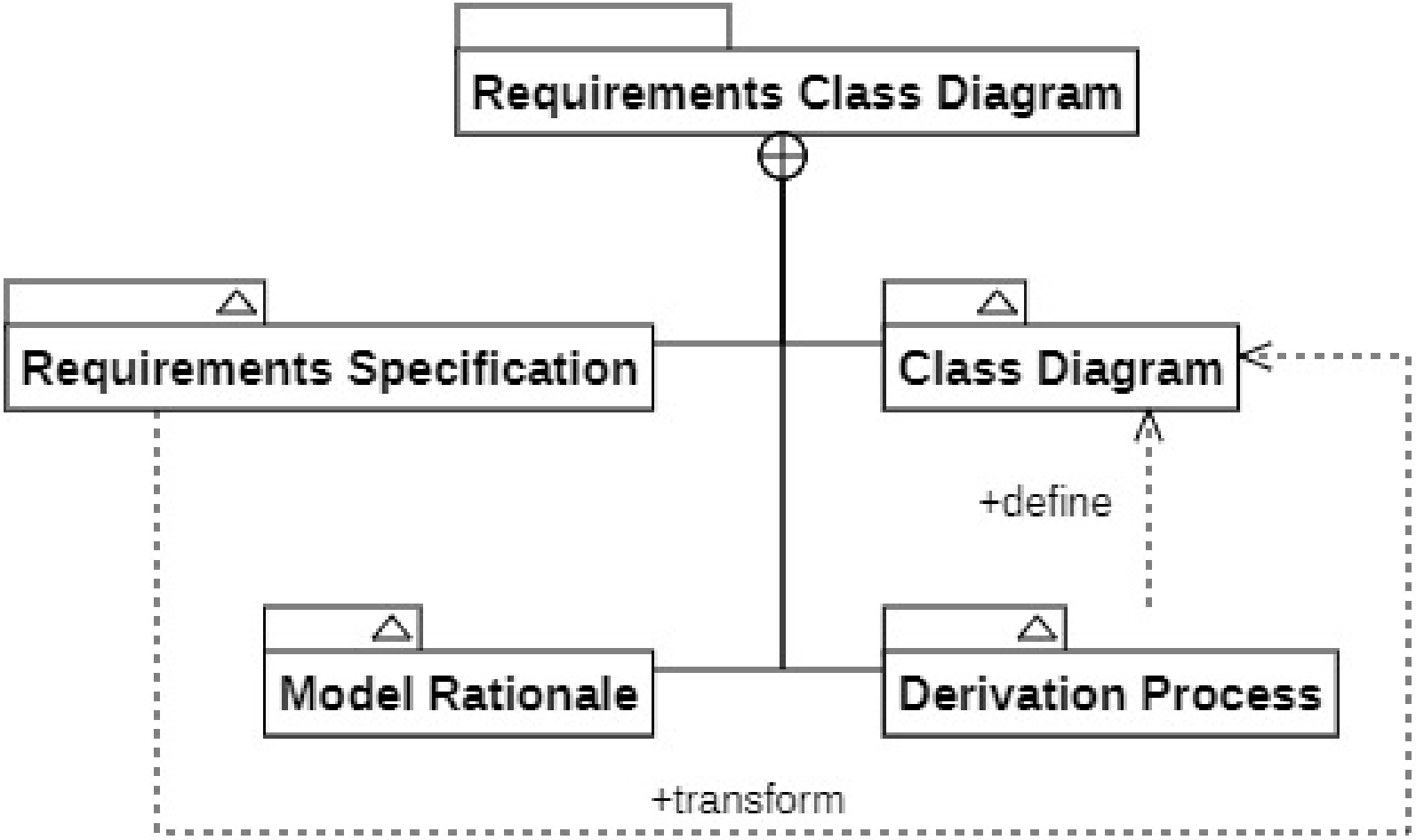
Reuse

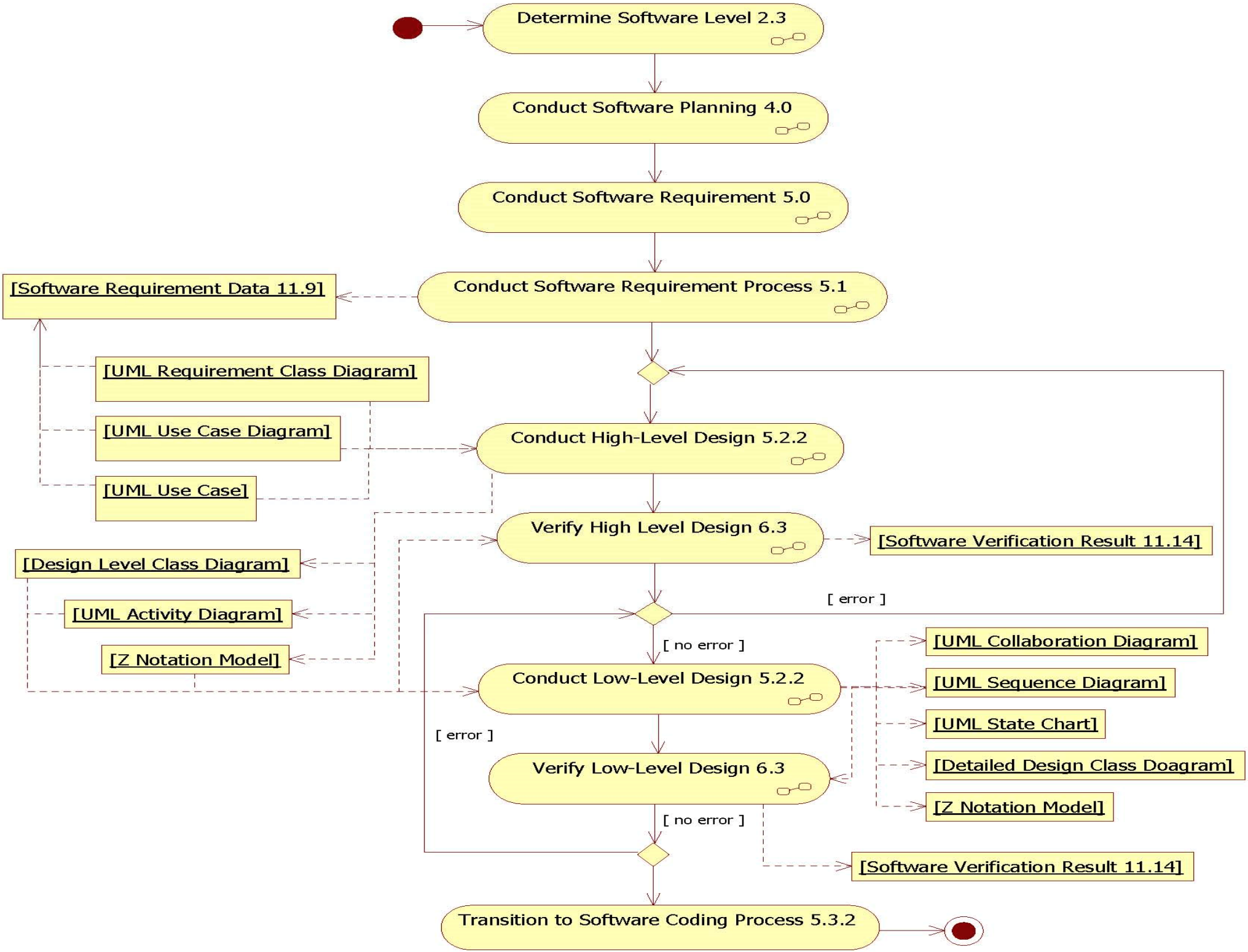

known requirements
(from sakeholders)
precise domain requirement
Specifications (to engineers)
Domain
Requirements
Engineering
feedback
Domain
Analysis and
Design
informal and
formal notations
feedback
feedback
domain models
(to engineers)
feedback
Application
Engineering
Domain
Language
Engineering
domain-specific
development rules
(from engineers)
application
requirements
application
models
executable
code
domain-specific
modeling language
(engineers' tool)
benchmark and evaluate
domain engineering

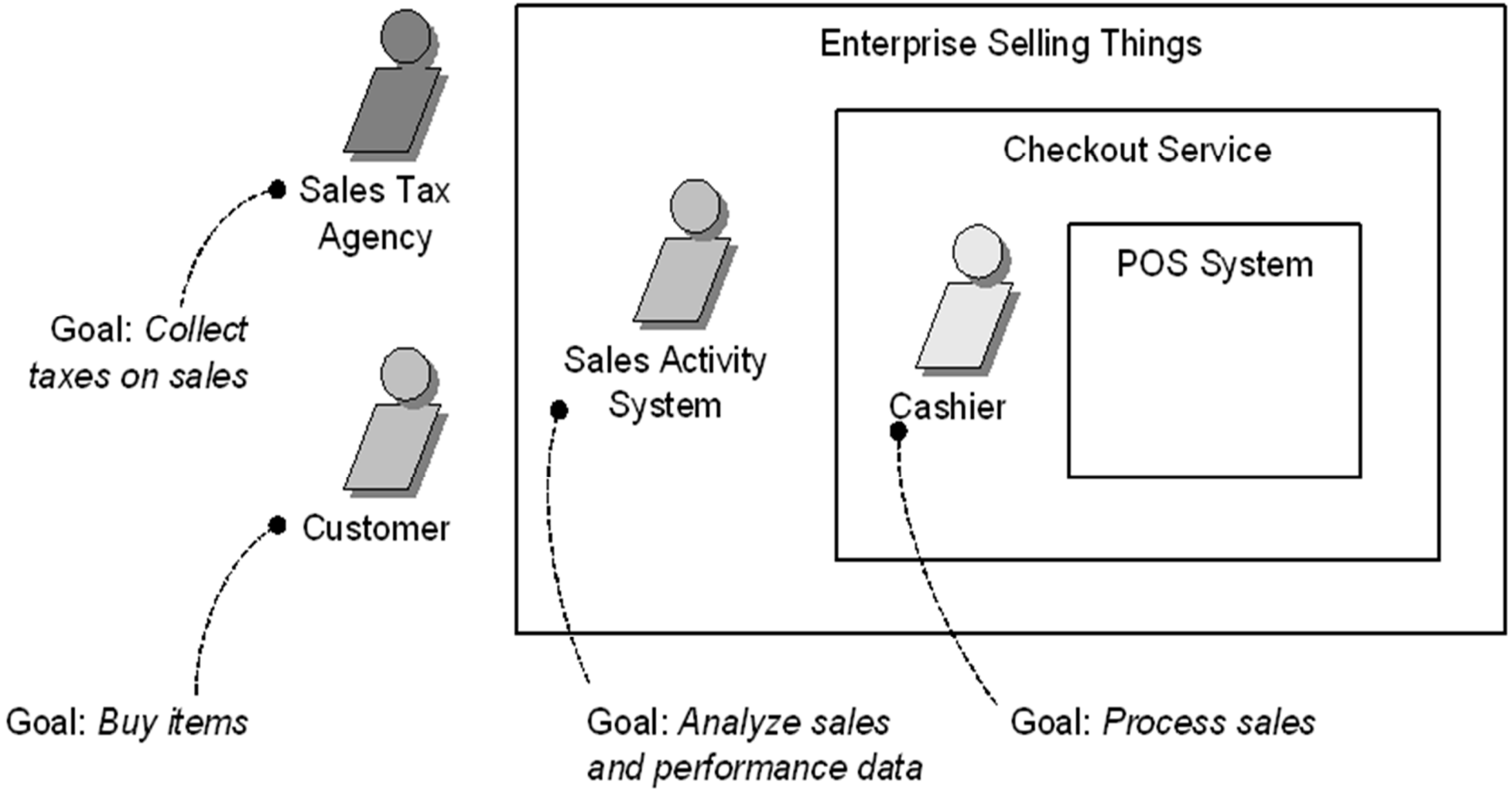
Model-Driven Software Engineering



Requirements Class Diagram




Determine Software Level 2.3
Conduct Software Planning 4.0
Conduct Software Requirement 5.0
Conduct Software Requirement Process 5.1
[Software Requirement Data 11.9]
[UML Requirement Class Diagram]
[UML Use Case Diagram]
[UML Use Case]
Conduct High-Level Design 5.2.2
Verify High Level Design 6.3
[Software Verification Result 11.14]
[Design Level Class Diagram]
[UML Activity Diagram]
[Z Notation Model]
[error]
[no error]
Conduct Low-Level Design 5.2.2
[UML Collaboration Diagram]
[UML Sequence Diagram]
[UML State Chart]
[Detailed Design Class Doagram]
[Z Notation Model]
[error]
Verify Low-Level Design 6.3
[no error]
[Software Verification Result 11.14]
Transition to Software Coding Process 5.3.2

OBJECT-ORIENTED MODELING PRINCIPLES

Operational Environment