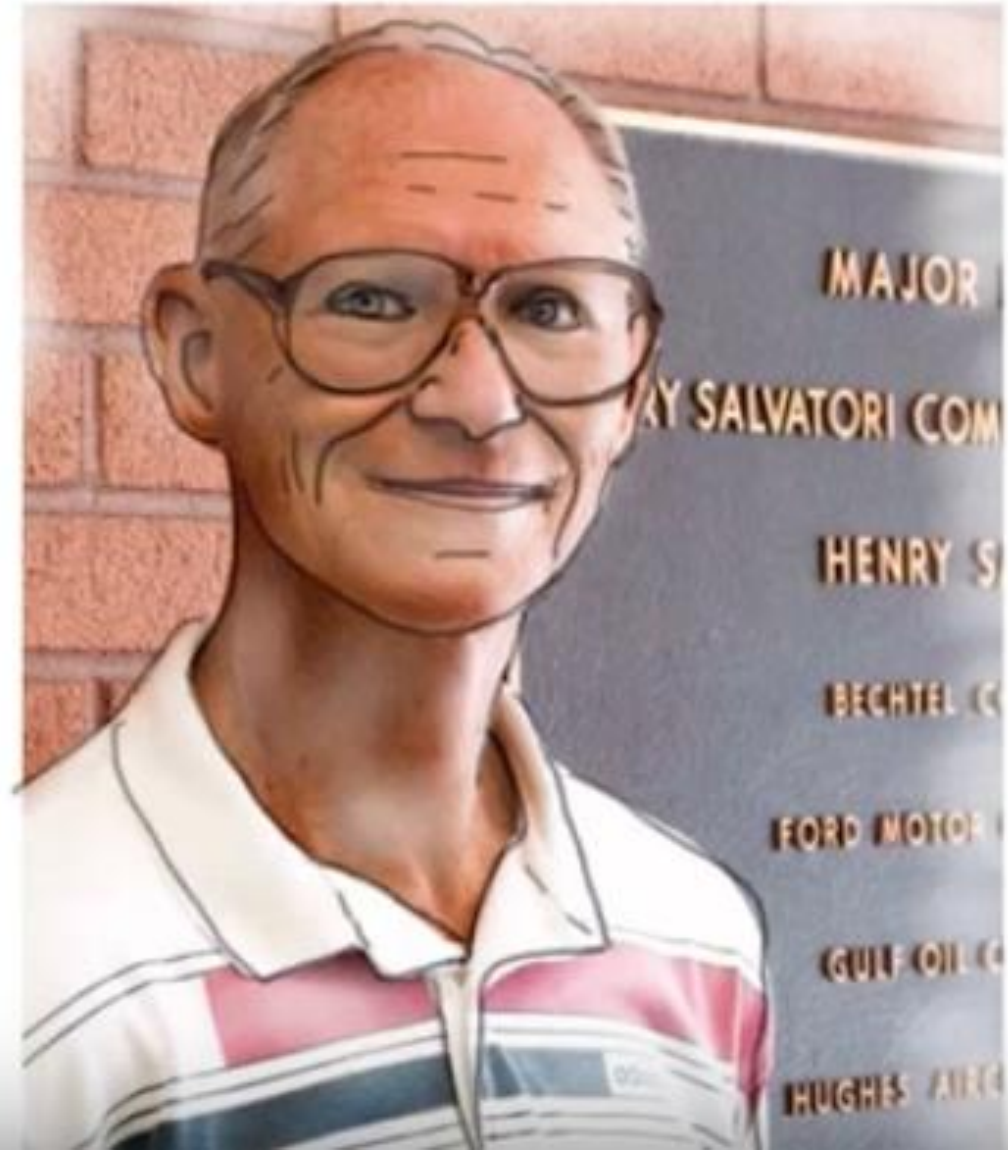
CSE3111 Software Engineering

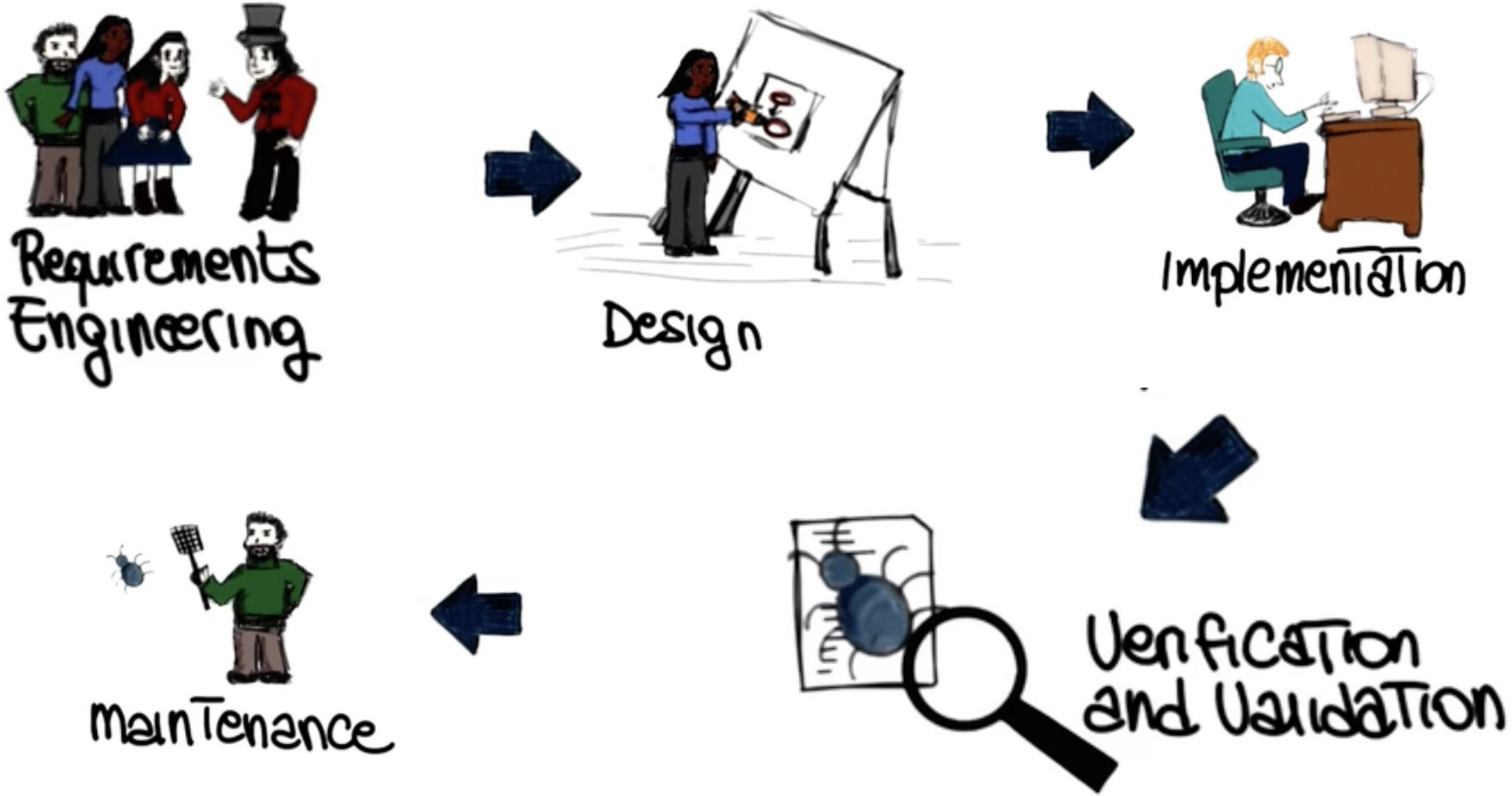
Life Cycle Model

BARRY BOHEM

BARRY BOHEM



Traditional Software Phases

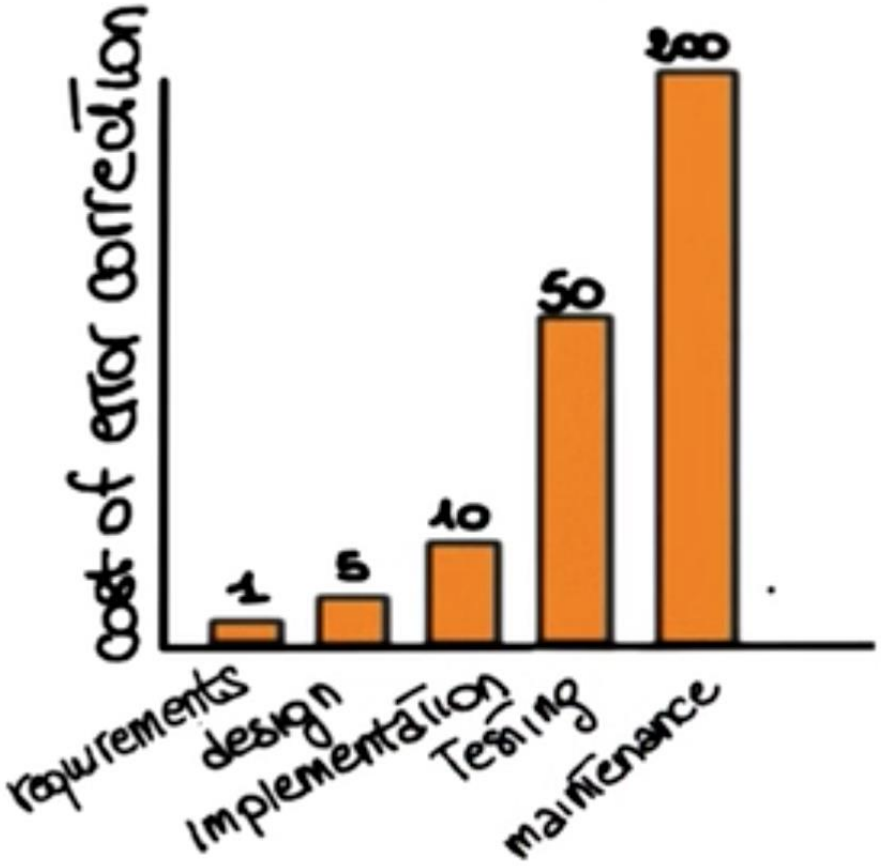


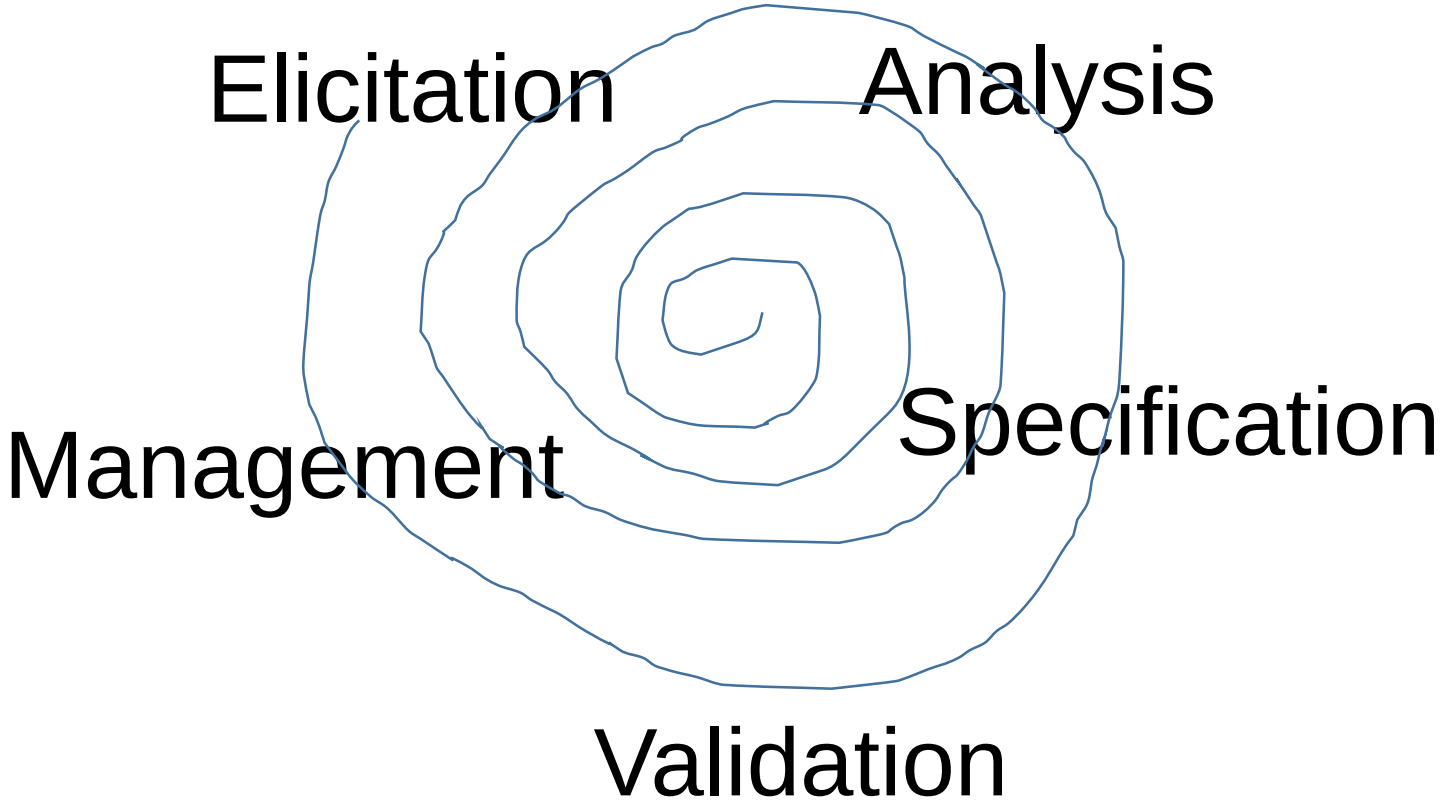
Requirements Engineering

which is the field within software engineering that deals with establishing the needs of stakeholders that are to be solved by the software.

Requirements Engineering

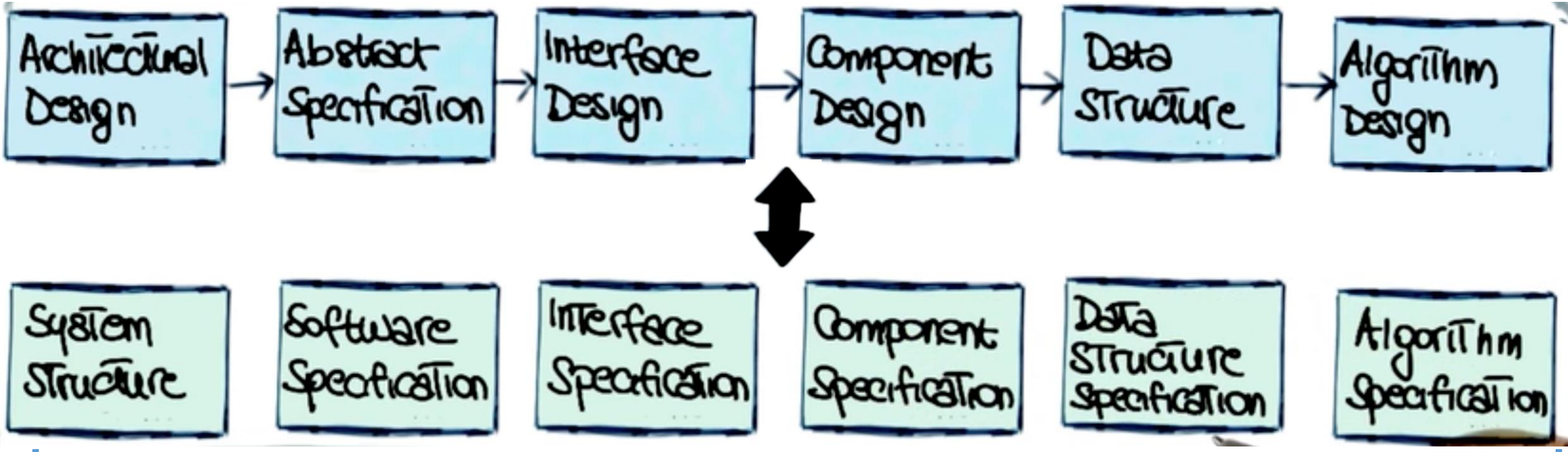
Cost of late correction





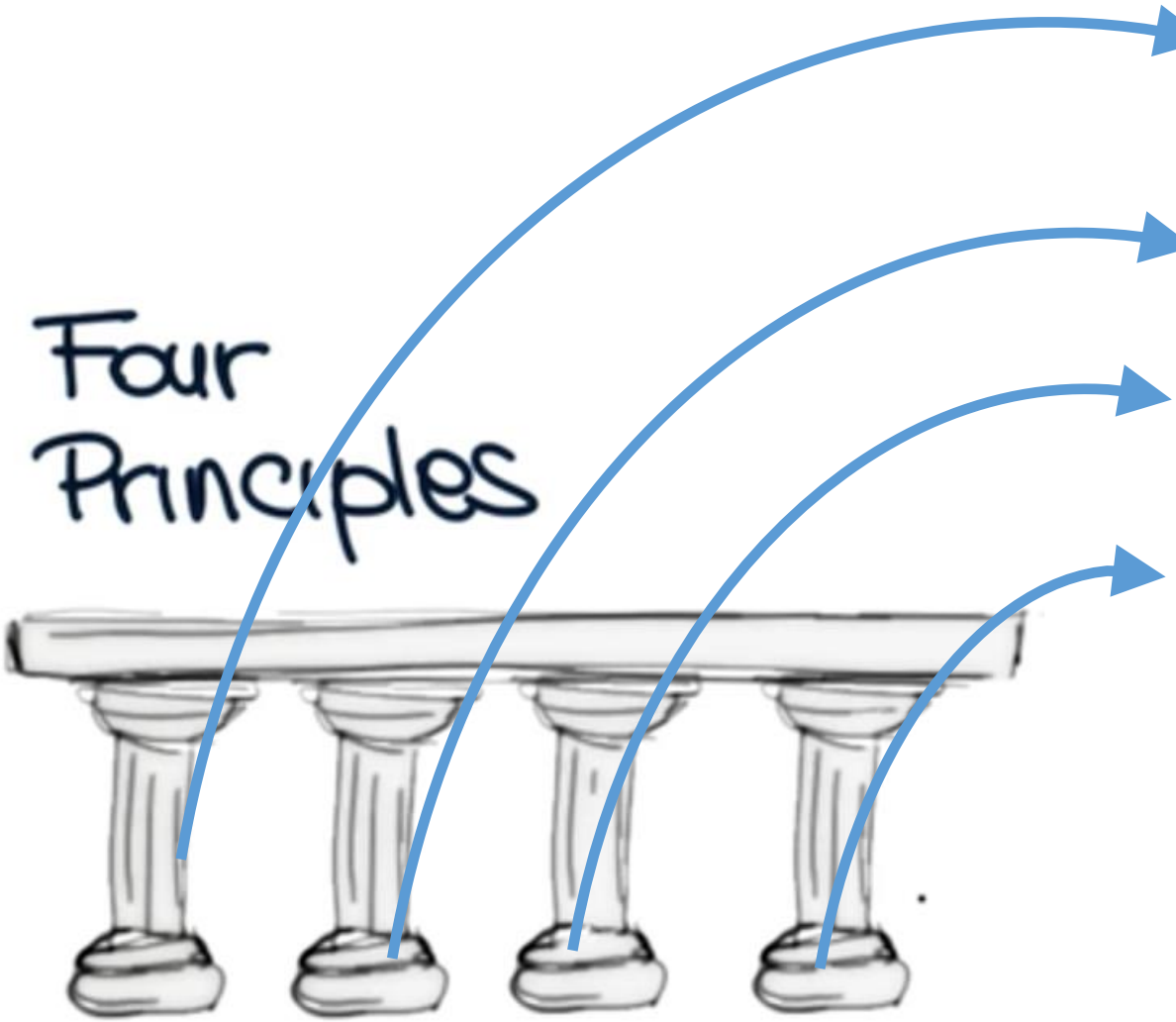
Design

Design Activities



Design Products

Implementation



Reduction of Complexity

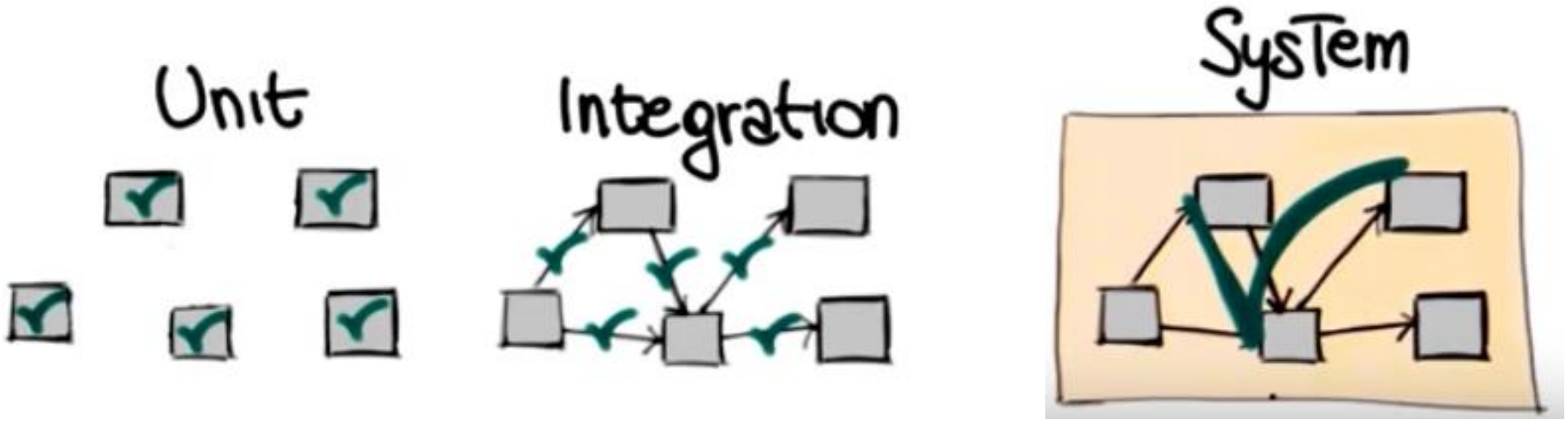
Anticipation of diversity

Structuring for validation

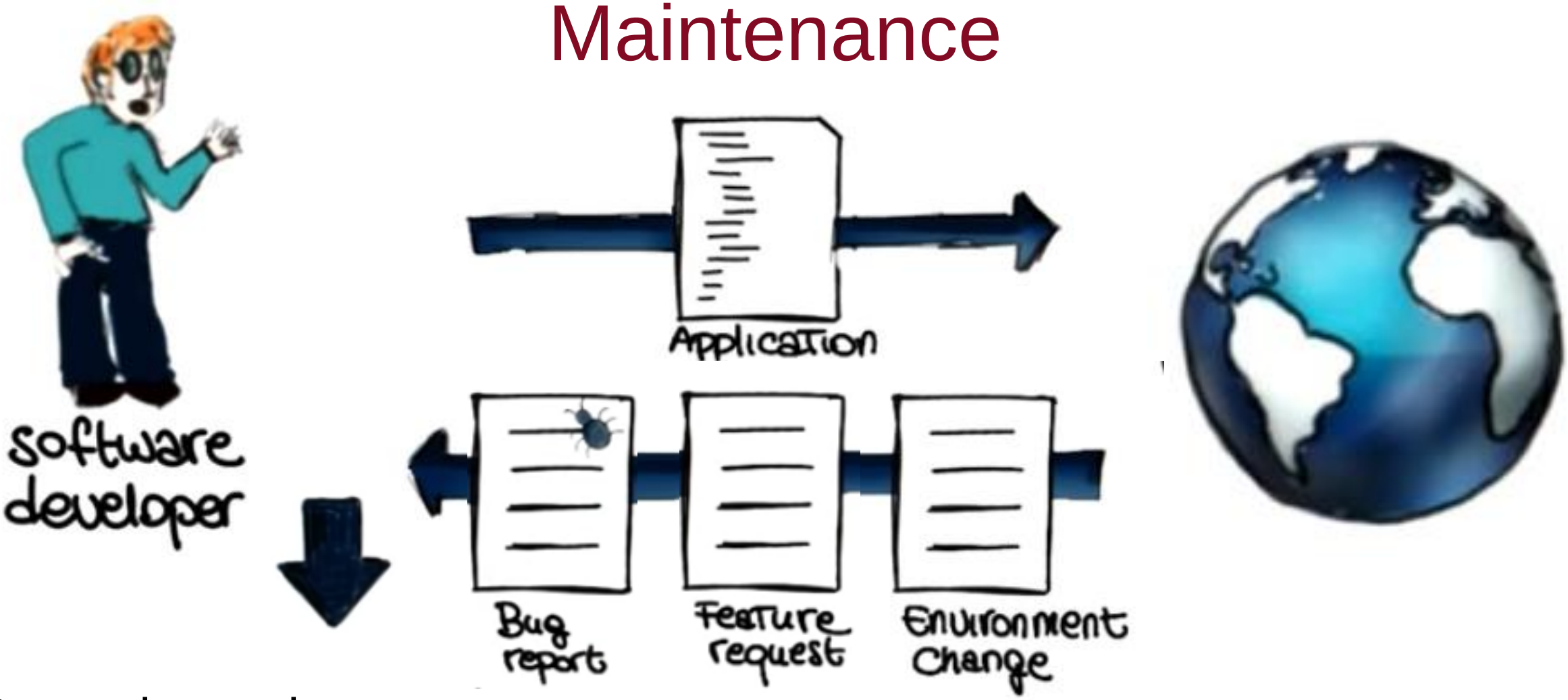
Use of (external) Standards

Verification & Validation

- Validation: did we build the right system?
- Verification: did we build the system right?



Maintenance



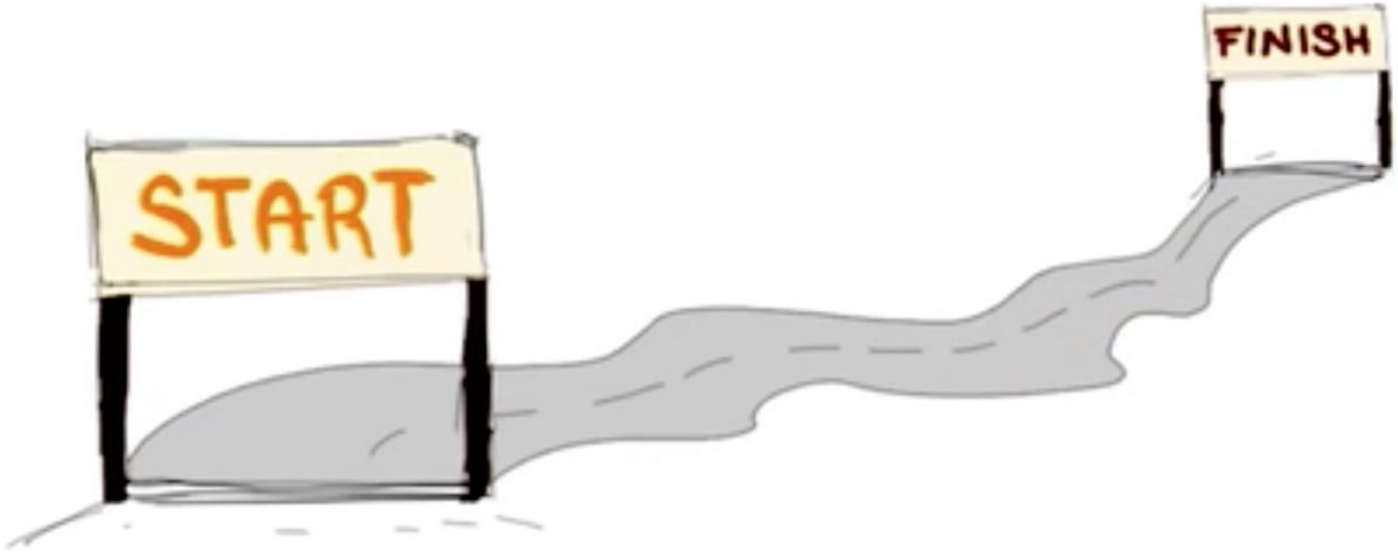
Corrective maintenance
Perfective maintenance
Adaptive maintenance

?

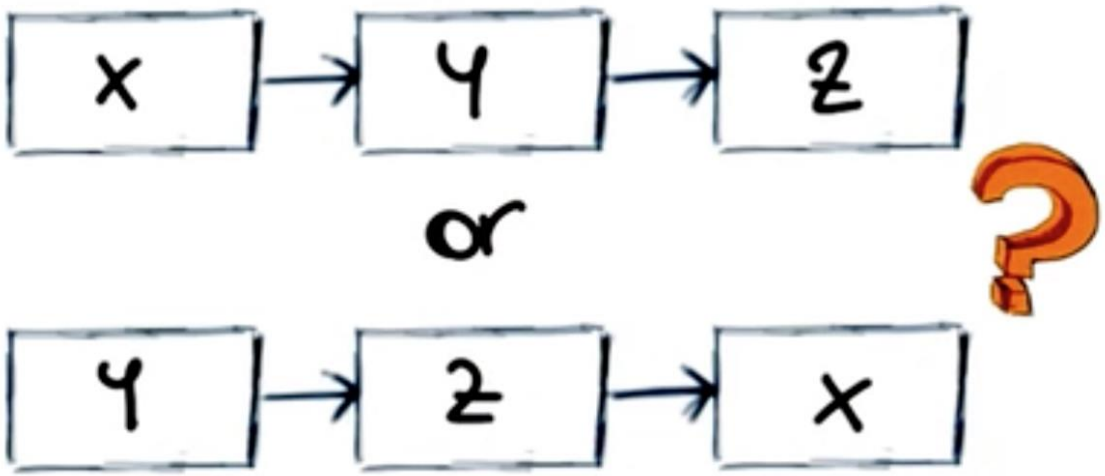
What are the traditional software phases?

- Requirements engineering, design, abstraction, implementation, verification & validation?
- Design, optimization, implementation, verification & validation, maintenance
- Requirements engineering, design, implementation, verification & validation, maintenance

Software Process/Lifecycle Model

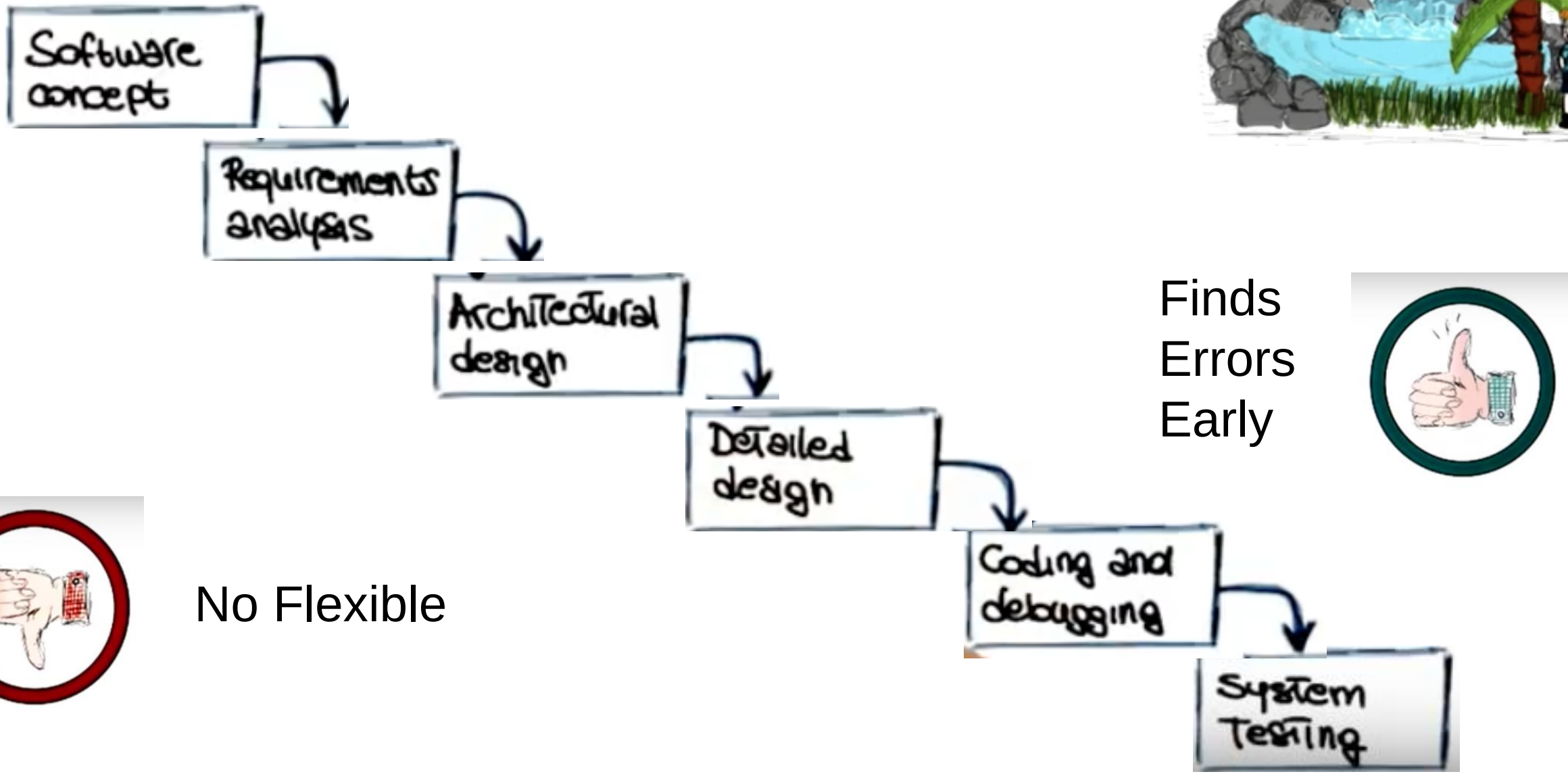


Determine the order



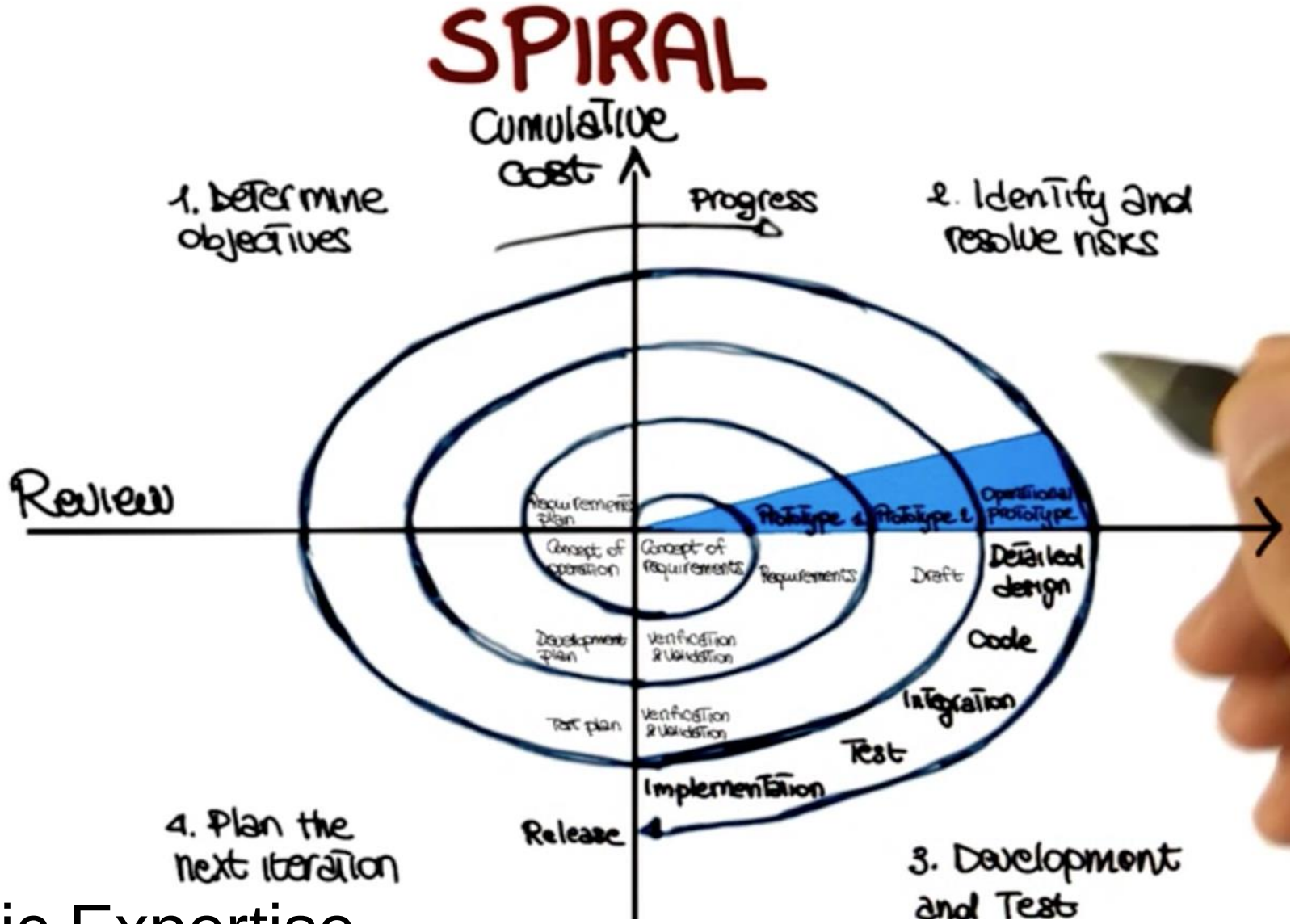
Establish The Transition criteria

Waterfall



Finds Errors Early

No Flexible



- Risk Reduction
- Functionality can be added
- Software Produces Early

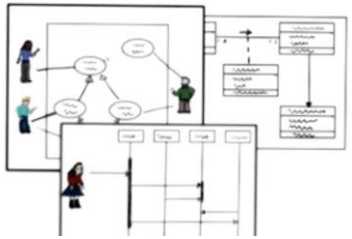
- Specific Expertise
- Highly Dependent on Risk Analysis
- Complex

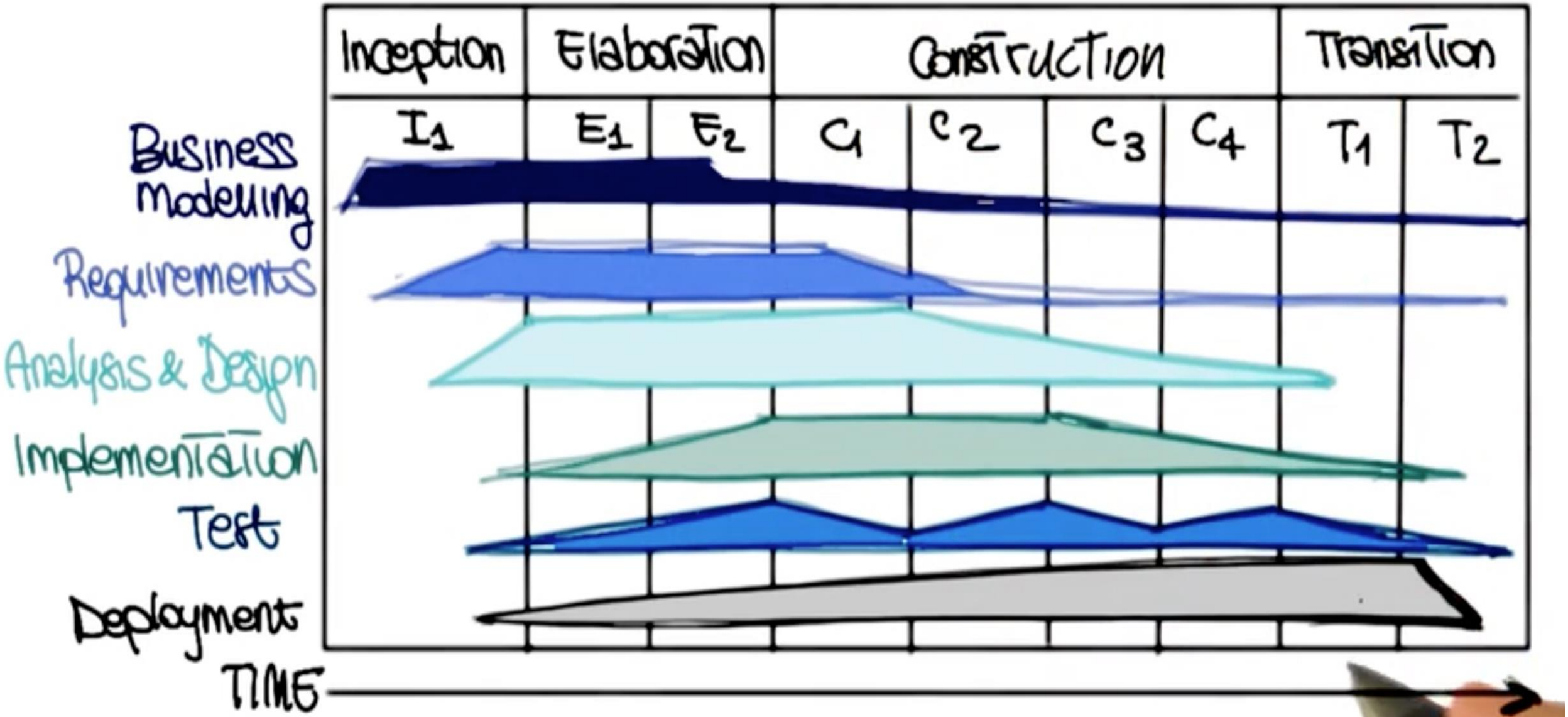
Evolutionary Prototyping

Immediate Feedback

Difficult to Plan

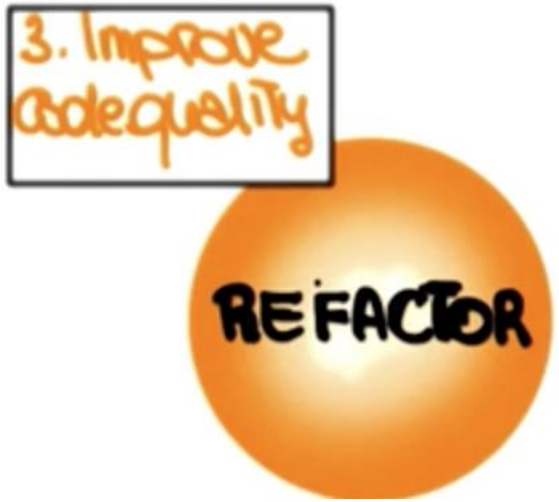
Rational Unified Process(RUP)





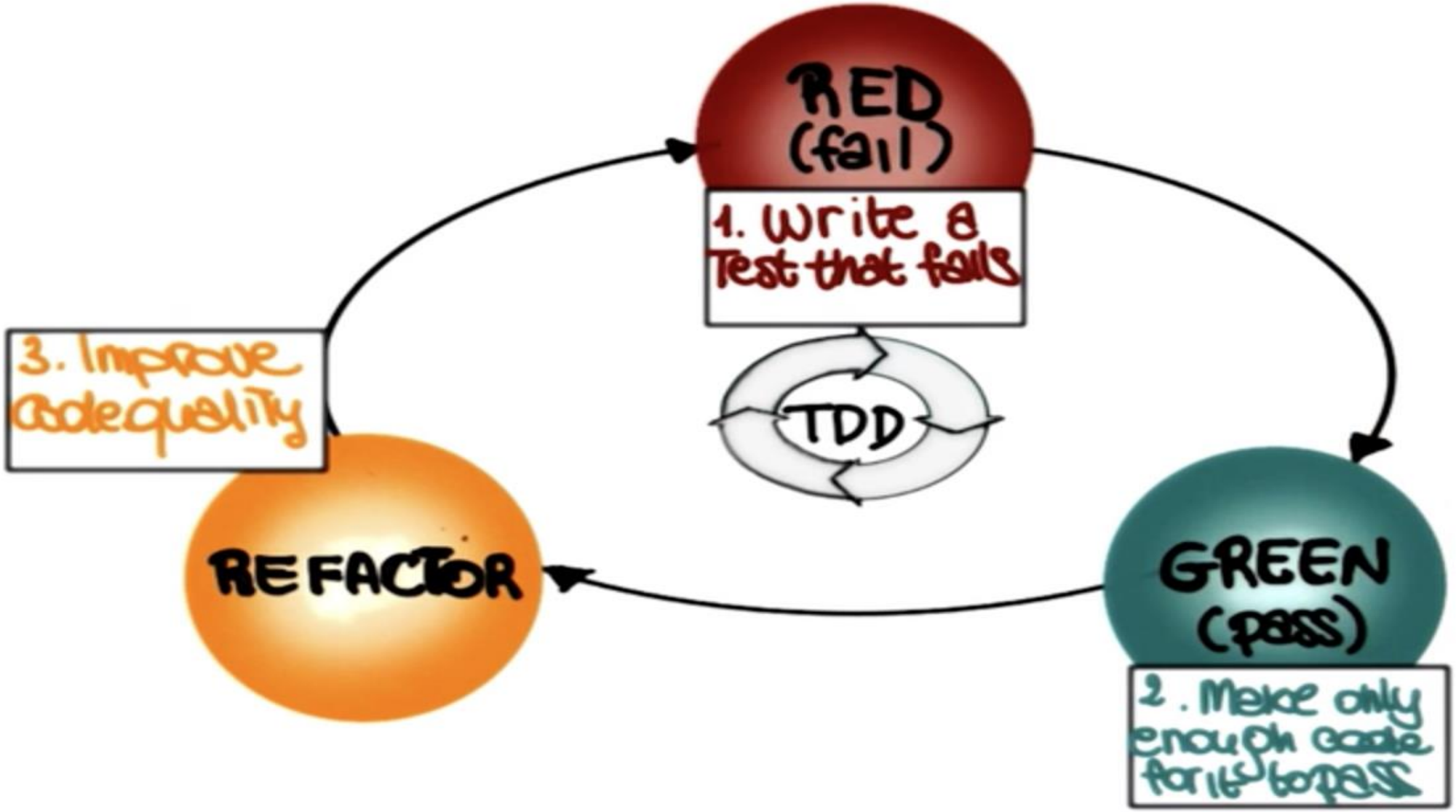
Agile


RED
(fail)
1. Write a
Test that fails
TDD


3. Improve
code quality
REFACTOR


GREEN
(pass)
2. Make only
enough code
for it to pass

Agile


RED
(fail)
1. Write a
Test that fails
TDD
GREEN
(pass)
2. Make only
enough code
for it to pass
REFACTOR
3. Improve
code quality

Choosing a Software Process Model

Requirements Understanding

Expected lifeTime

Risk

Schedule Constraints

Interaction with management / customers

Expertise

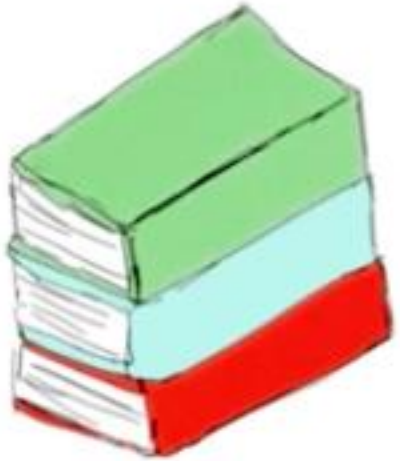
Which of the following models is most suitable to develop a software control system?

- [ ] Pure waterfall
- [ ] TDD
- [ ] Evolutionary prototyping

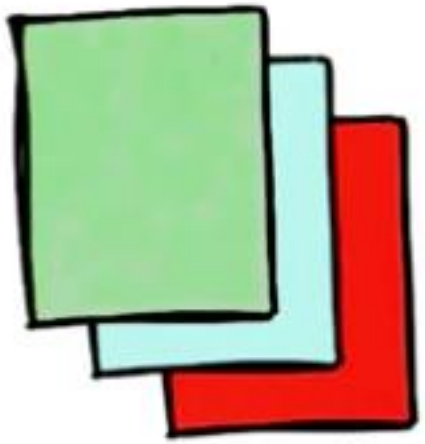
Which model is the most suitable if you expect midcourse corrections?

- [ ] Pure waterfall
- [ ] Spiral
- [ ] Evolutionary prototyping

Lifecycle Documents



IEEE Documents Definition



Light-weight documents

Classic Mistakes: People

Heroics

Work environment

People management

Classic Mistakes: Process

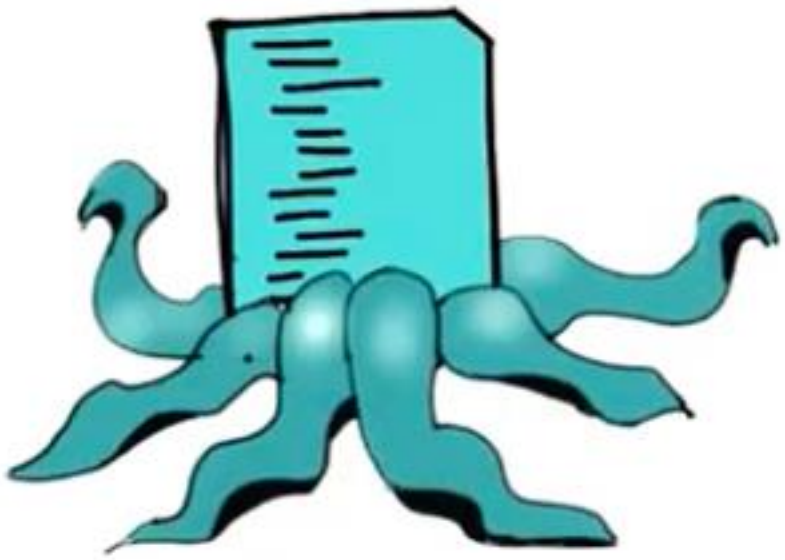
Scheduling issues

Planning issues

Failures

Classic Mistakes: Product

Gold plating



Feature creep

Research ≠ Development

Classic Mistakes: Technology

silver-bullet syndrome

switching tools

No person control

Which kind of mistake "adding people to a late project" is ?

[ ] people mistake
[ ] product mistake
[ ] technology mistake
[ ] it is not a mistake