Evolution & Maintenance

Software Engineering 2

(3103313-1)

Amirkabir University of Technology

Fall 1399-1400

Software Evolution

Software systems continually adapt and evolve during their lifetime.

Cost of Maintenance

Evolution vs. Maintenance

Evolution & Maintenance

- Long Lifetime
  - Large systems e.g., military and air traffic control systems
  - Costly enterprise software, many years to get an ROI.
  - Successful Software

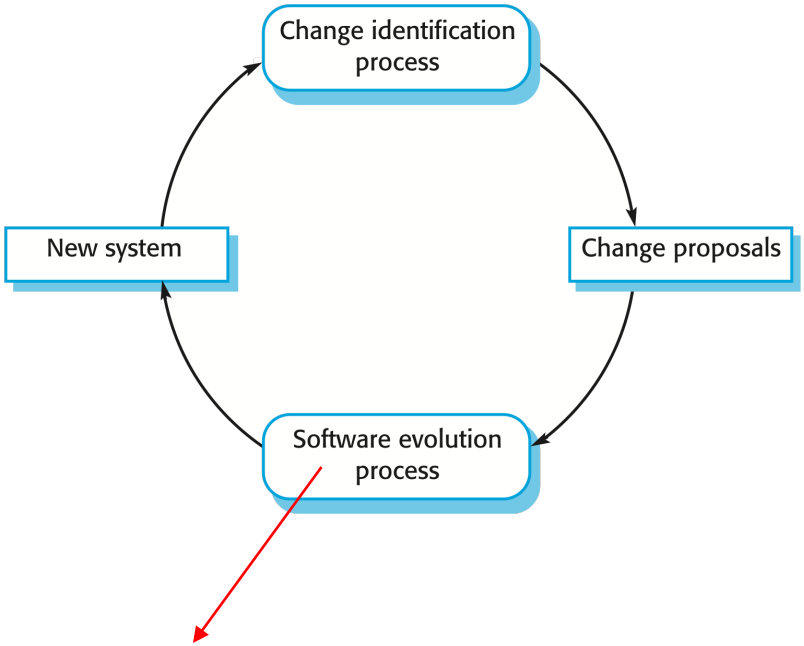
- Historical data suggests that somewhere between 60% and 90% of software costs are evolution costs.

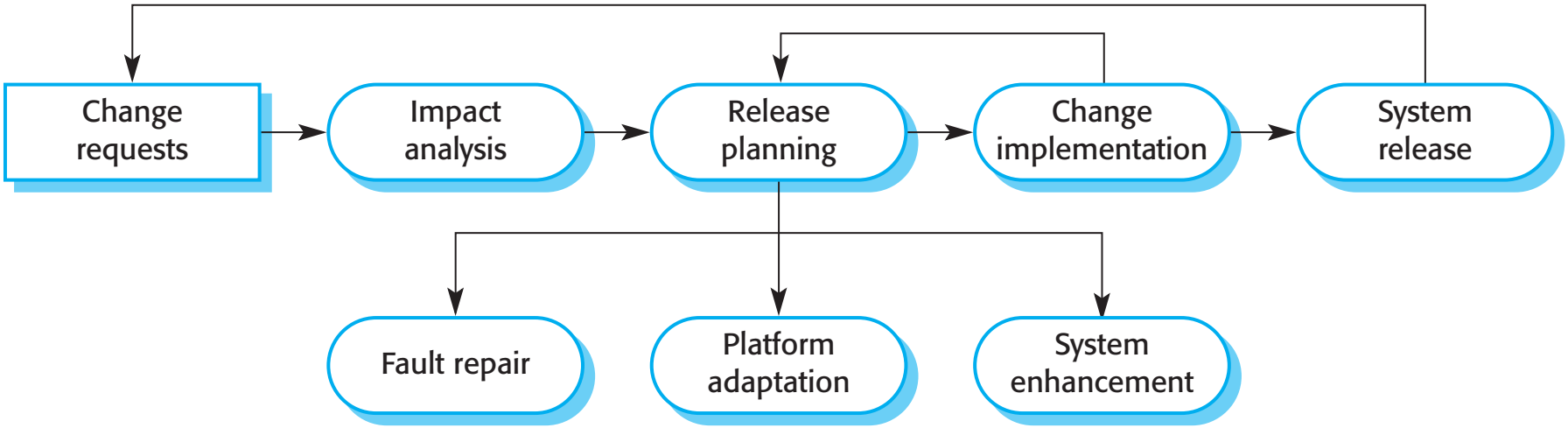
- Evolution vs. Maintenance
  - Seamlessness vs. Discontinuity

Evolution Process

Depends on

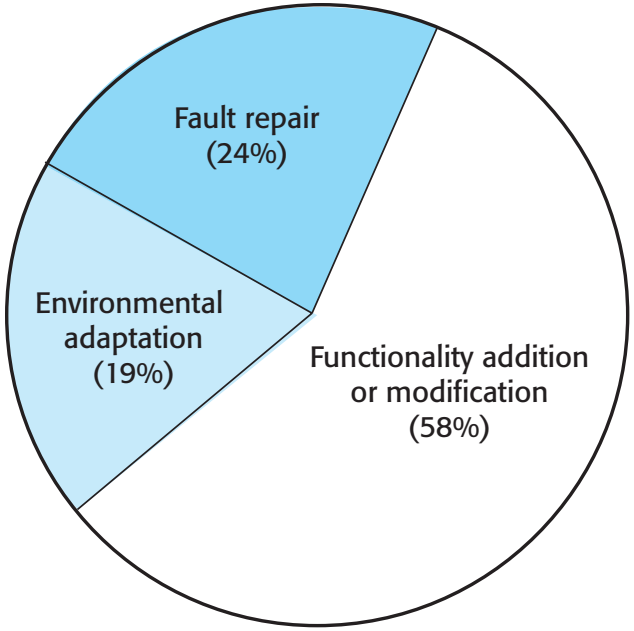
- Type of software (formal or informal)
- The development process
- The skill of the people involved





Types of Software Maintenance

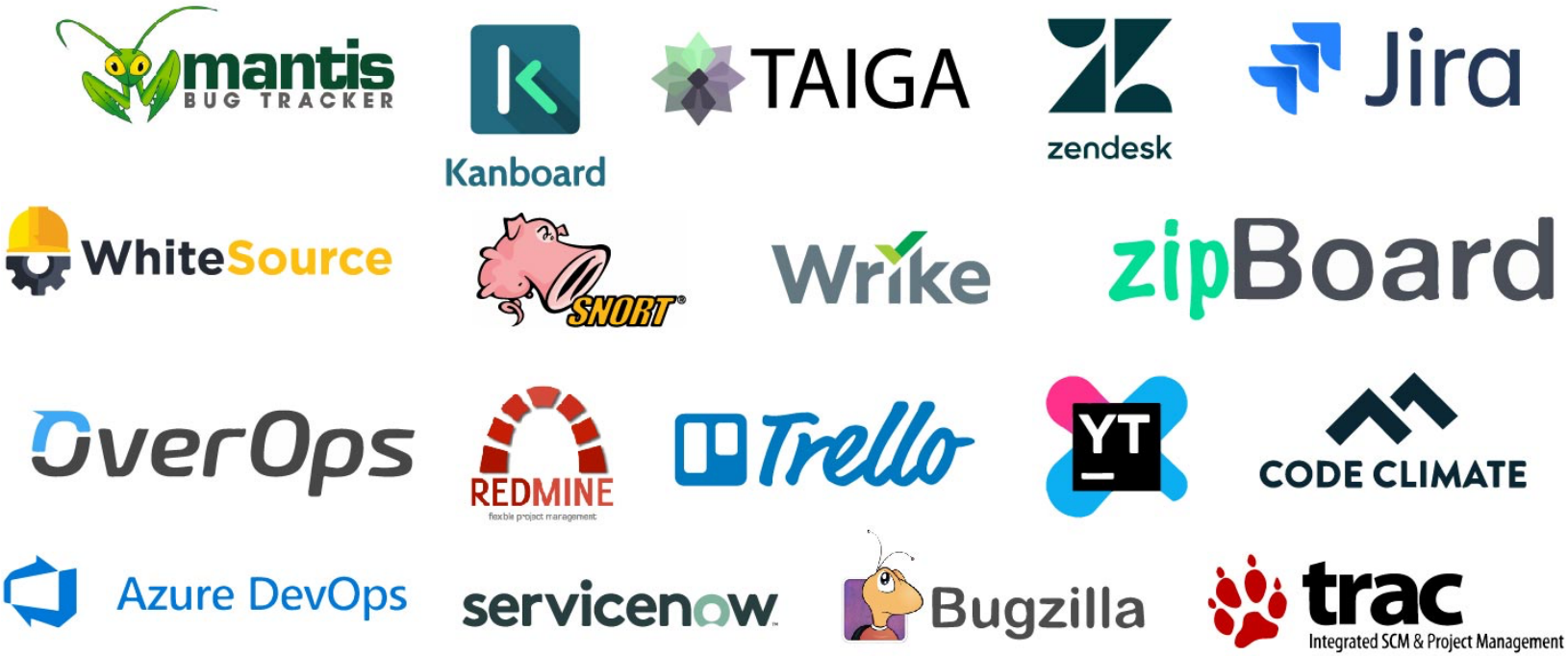
1. Fault repairs to fix bugs and vulnerabilities (Corrective)
2. Environmental adaptation to adapt the software to new platforms and environments (Adaptive)
3. Functionality addition to add new features and to support new requirements (Perfective)



4. Rewriting code/Refactoring (Preventive)

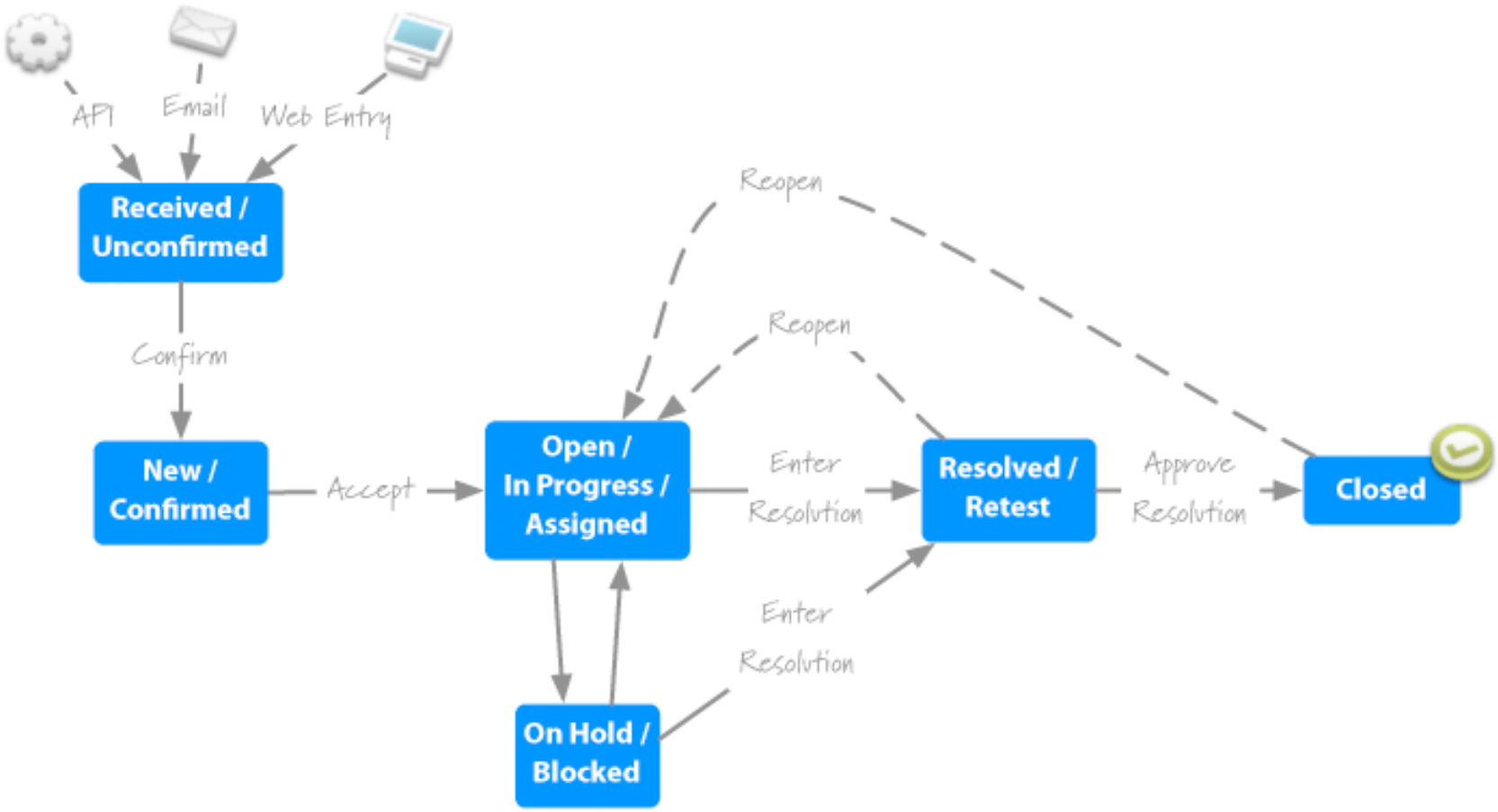
Issue Tracking Systems

Bug Tracking Systems, Defect Tracking Systems



Issue Tracking Systems

Bug Tracking Systems, Defect Tracking Systems

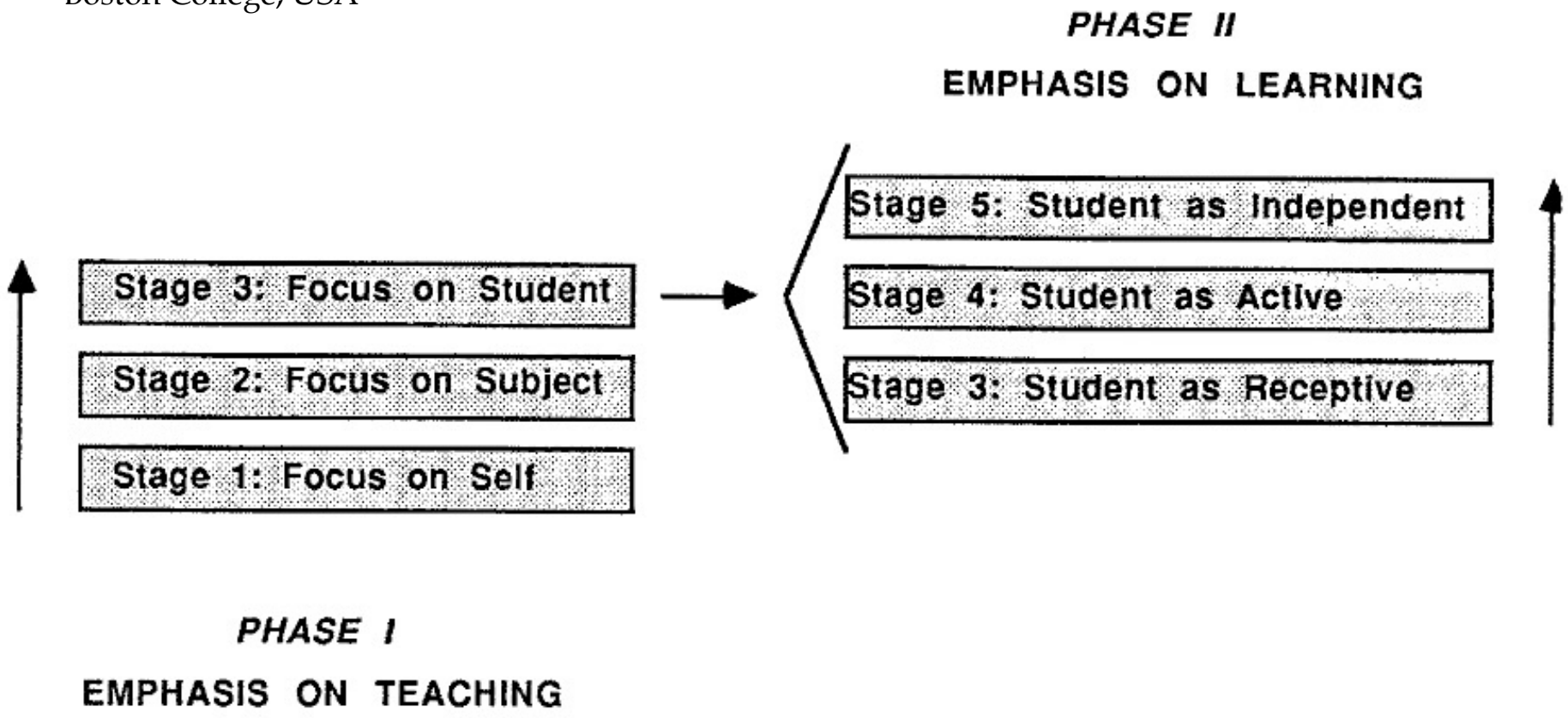


break

Life After Death by PowerPoint

How Professors Develop as Teachers

PETER KUGEL
Boston College, USA



Reengineering

Business Process Reengineering

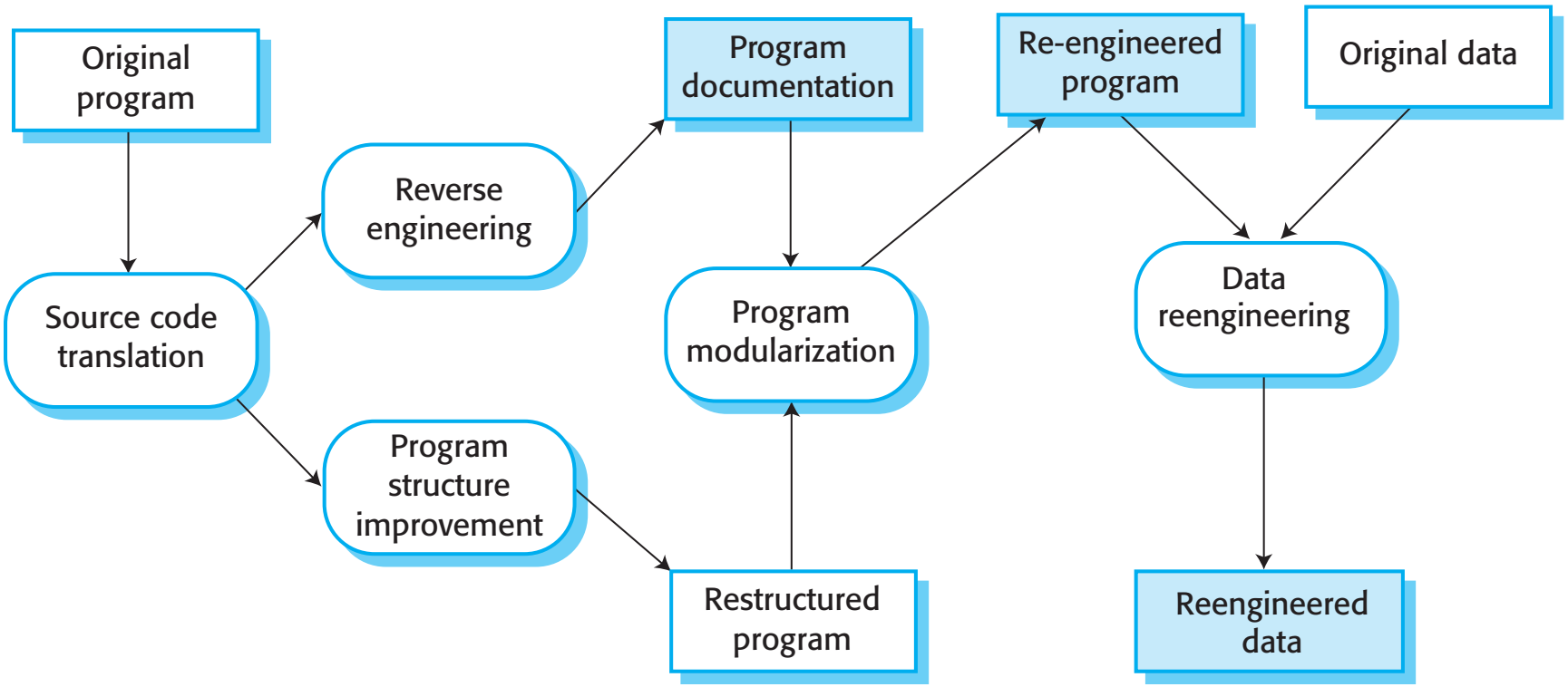
Software Reengineering

Reverse Engineering

Refactoring

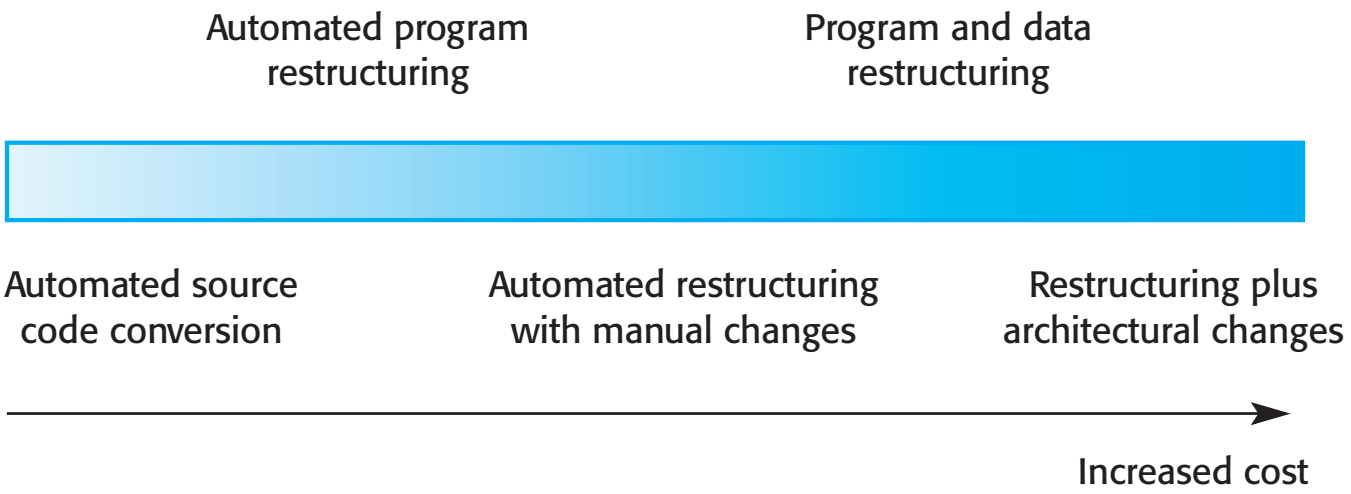
Software Reengineering

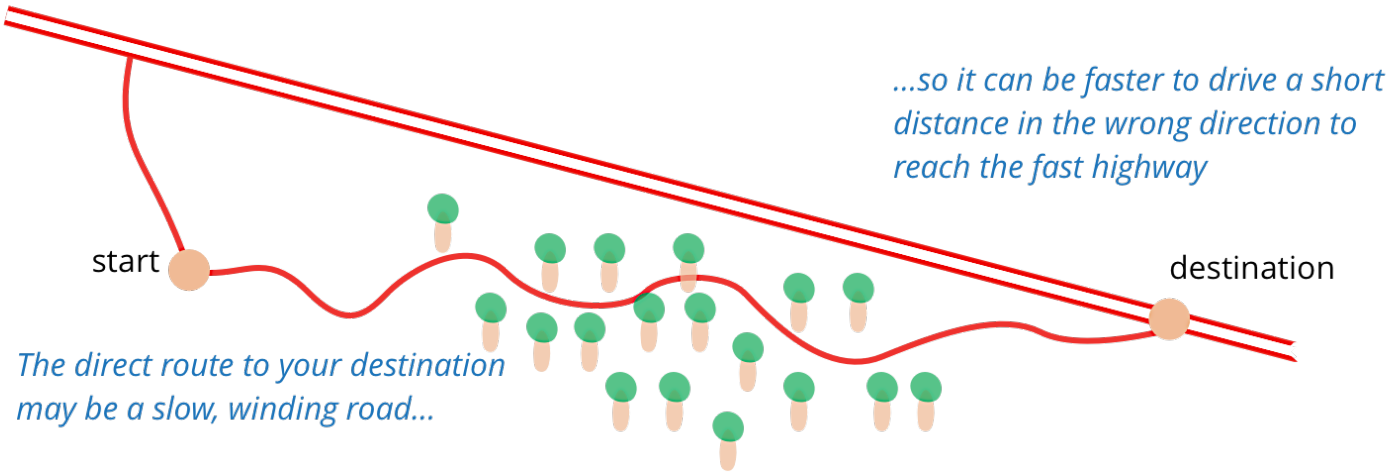
A general process model for legacy systems



Software Reengineering

Cost





Refactoring

Making the program amenable to future changes.

- improve the structure,
- reduce the complexity,
- make it easier to understand
- ...

Code Refactoring

- ✓ Clarification
- ✓ Reuse
- ✓ Improve Flexibility
- ✓ Bug Swarms
- ✓ Bad Programming Practice (“Bad Smells”)
  - Duplicate code, Long methods, Switch (case) statements, Data clumping, Speculative generality, …

- ✘ Individual Bugs
- ✘ Not Invented Here

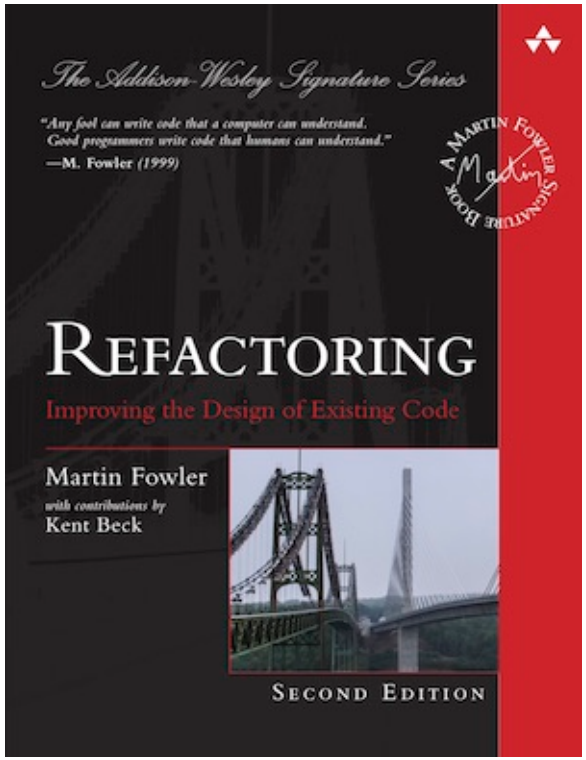
Code Refactoring

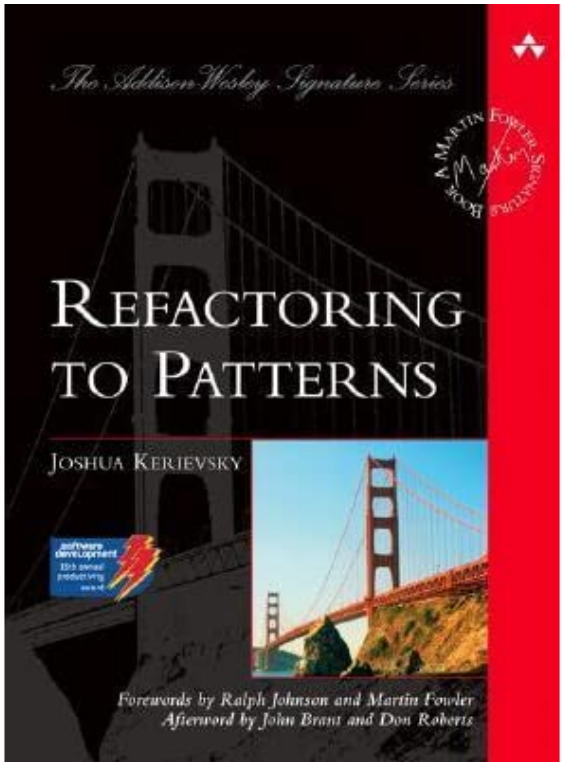
- ✓ Clarification
- ✓ Reuse
- ✓ Improve Flexibility
- ✓ Bug Swarms
- ✓ Bad Programming Practice (“Bad Smells”)
  - Duplicate code, Long methods, Switch (case) statements, Data
    Speculative generality, …

Eclipse (JDT)
UCDetector
AutoRefactor
…

- ✘ Individual Bugs
- ✘ Not Invented Here

Refactoring





https://refactoring.com/

... last but not least!

Economics of Software Maintenance