

# Software Life-Cycles

# Software life cycle

✍ Life cycle models

✍ Development strategies

✍ Choosing a life cycle model

# Software development

Idea

Requirement

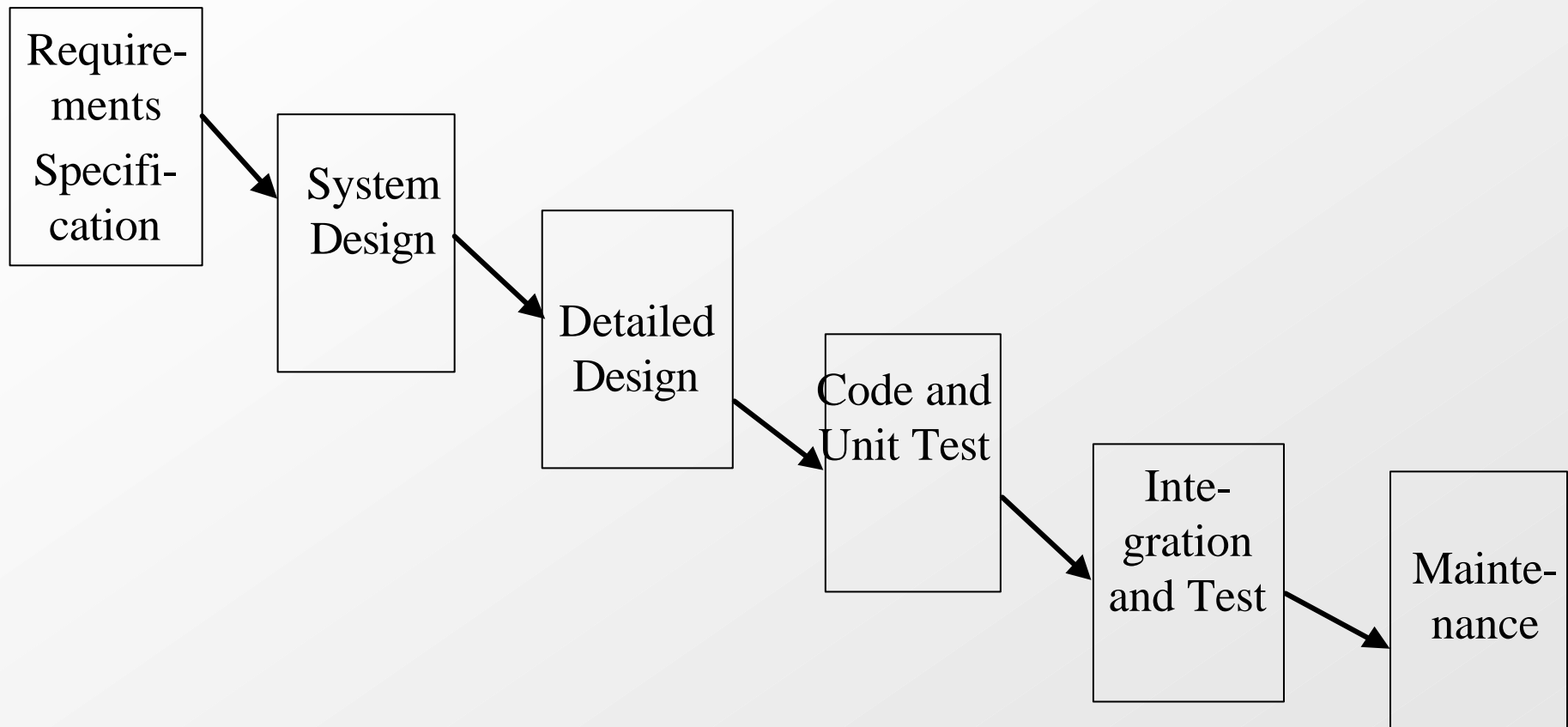
Create system

Use system

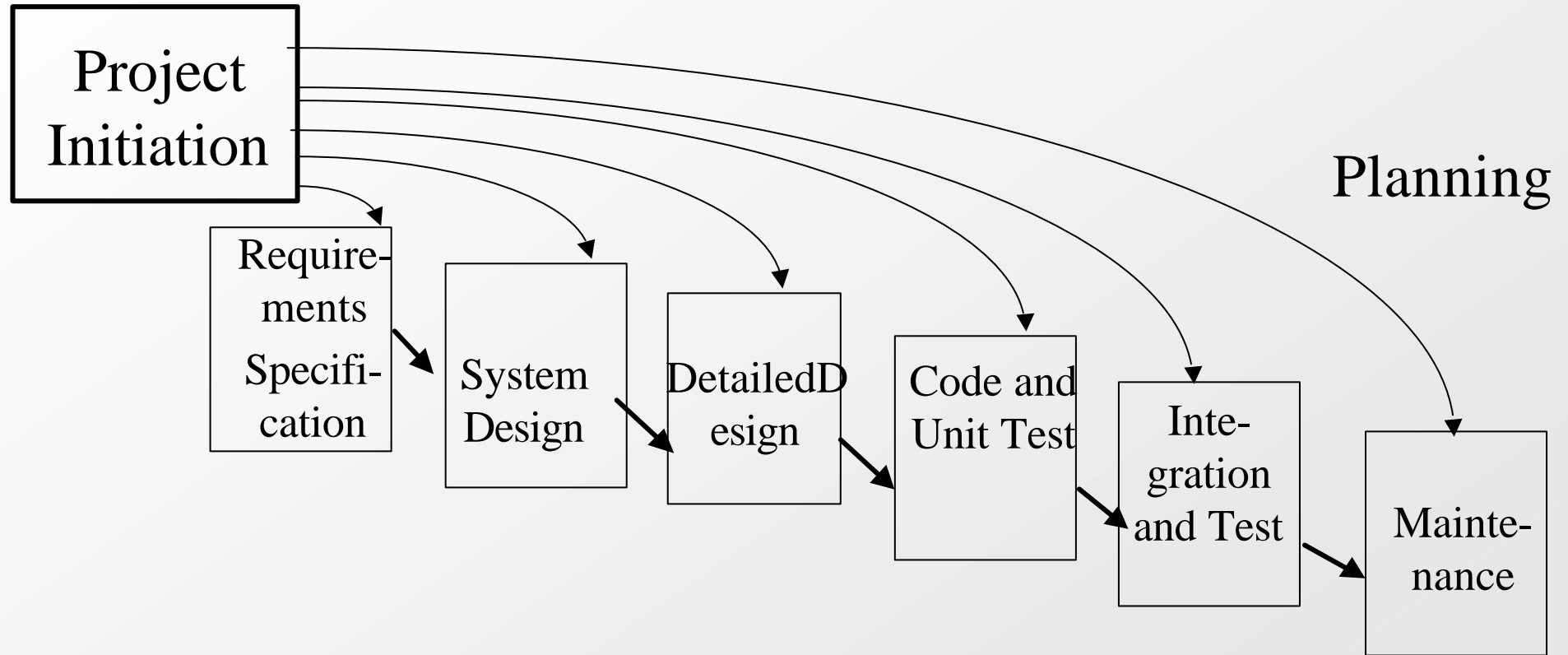
Must be formalised

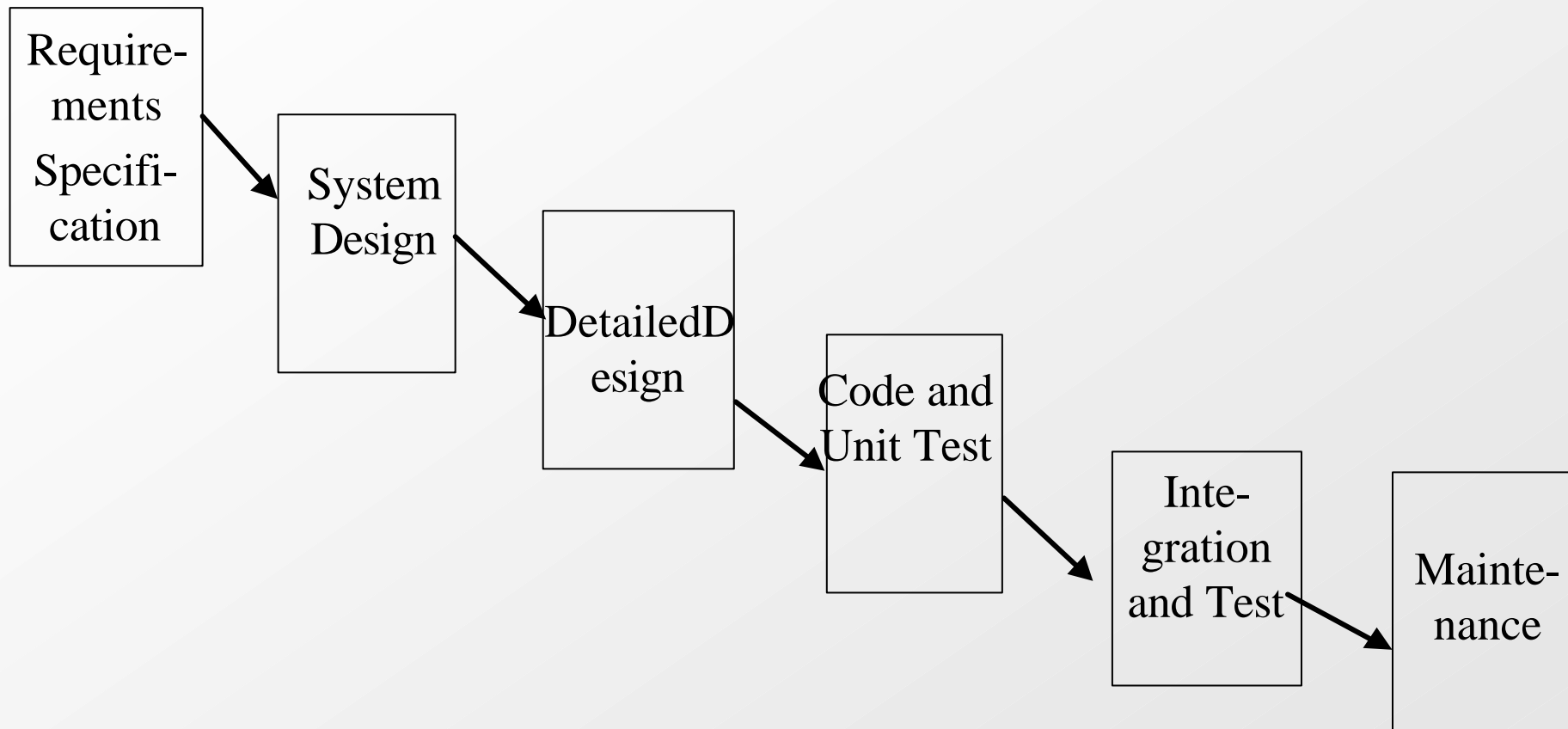
= LIFE CYCLE MODEL

# Life cycle model waterfall

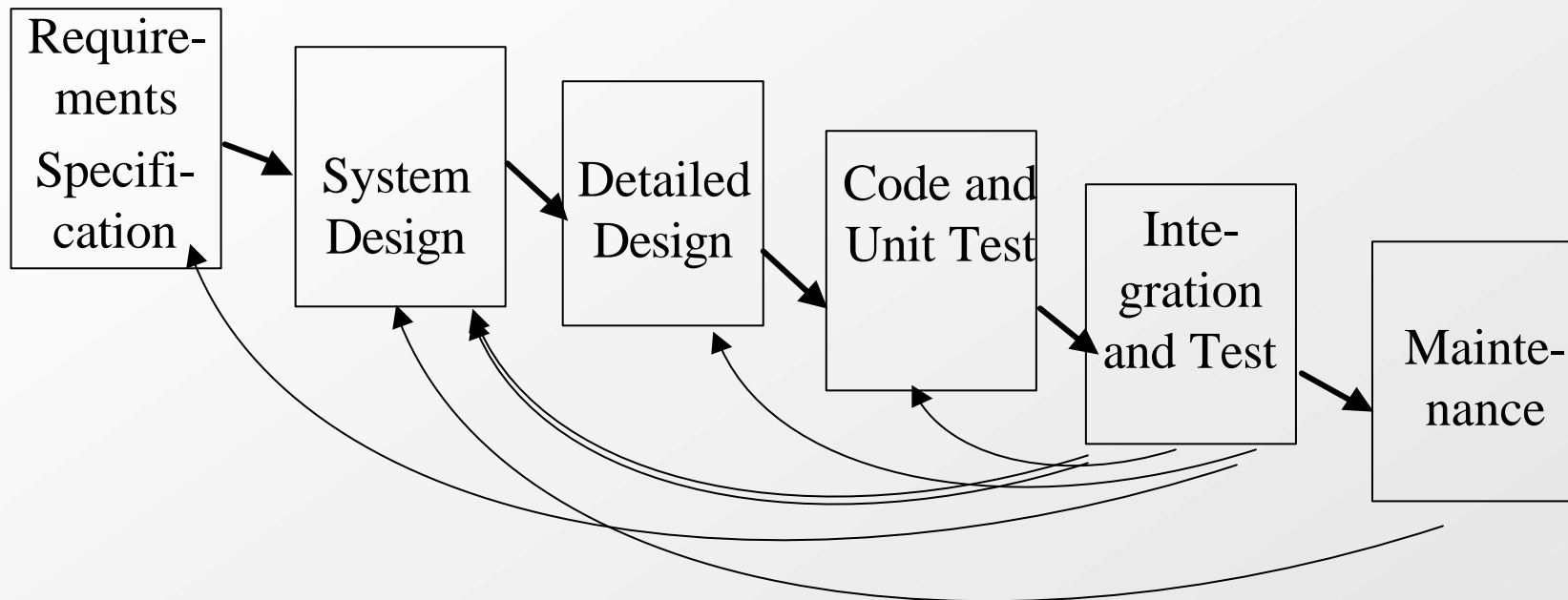


# Planning



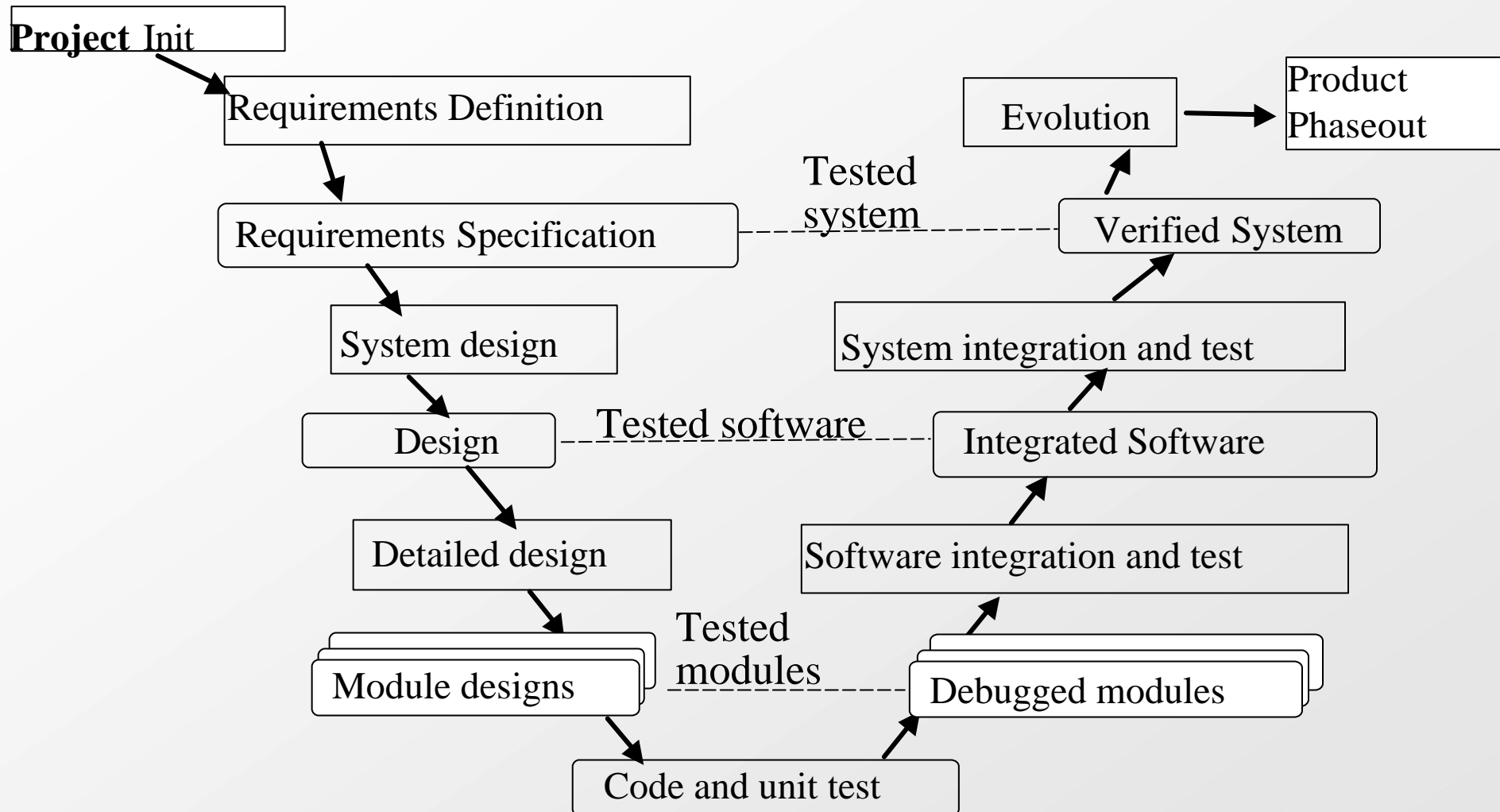


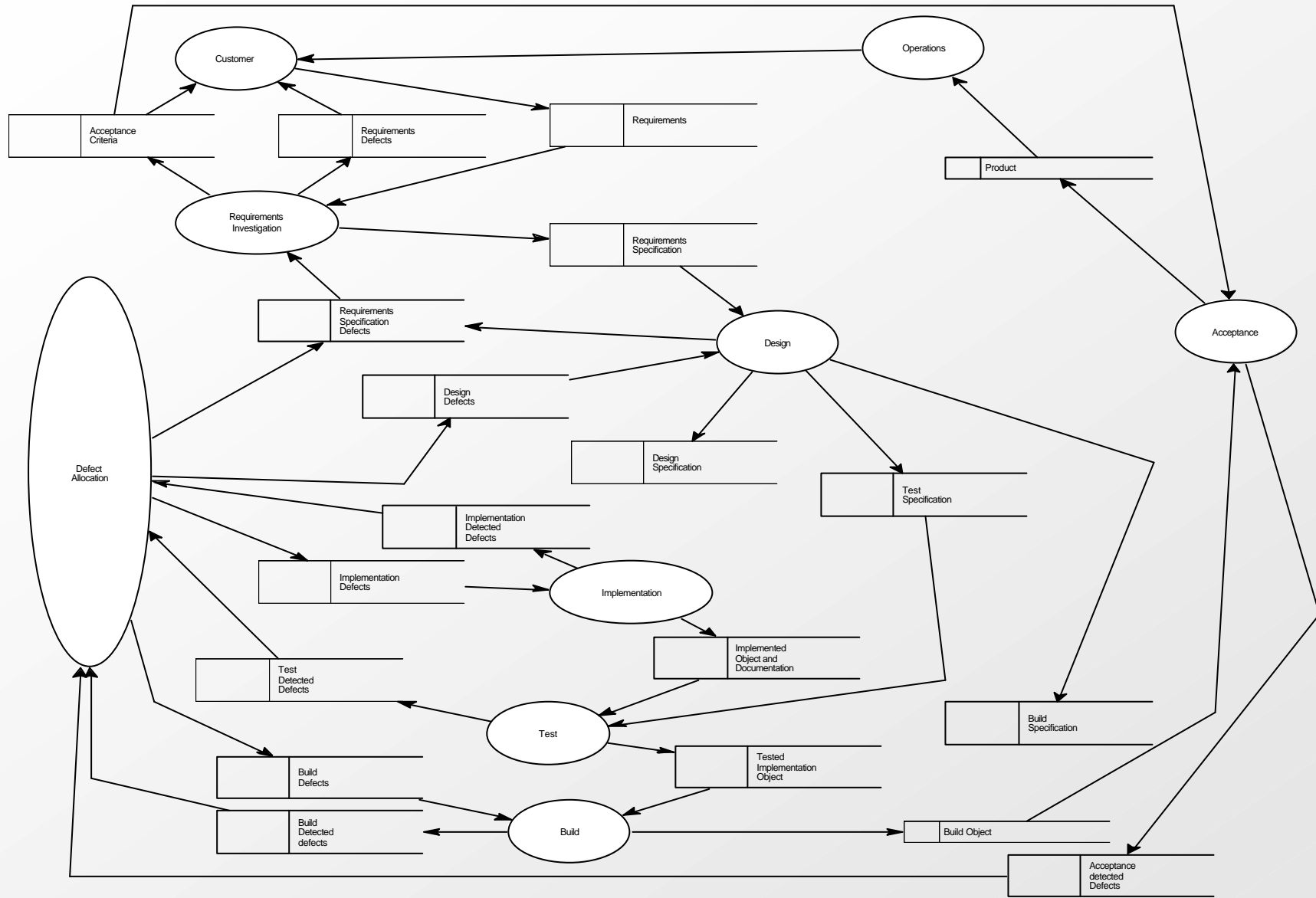
# Feedback



Feedback

# The “V” model

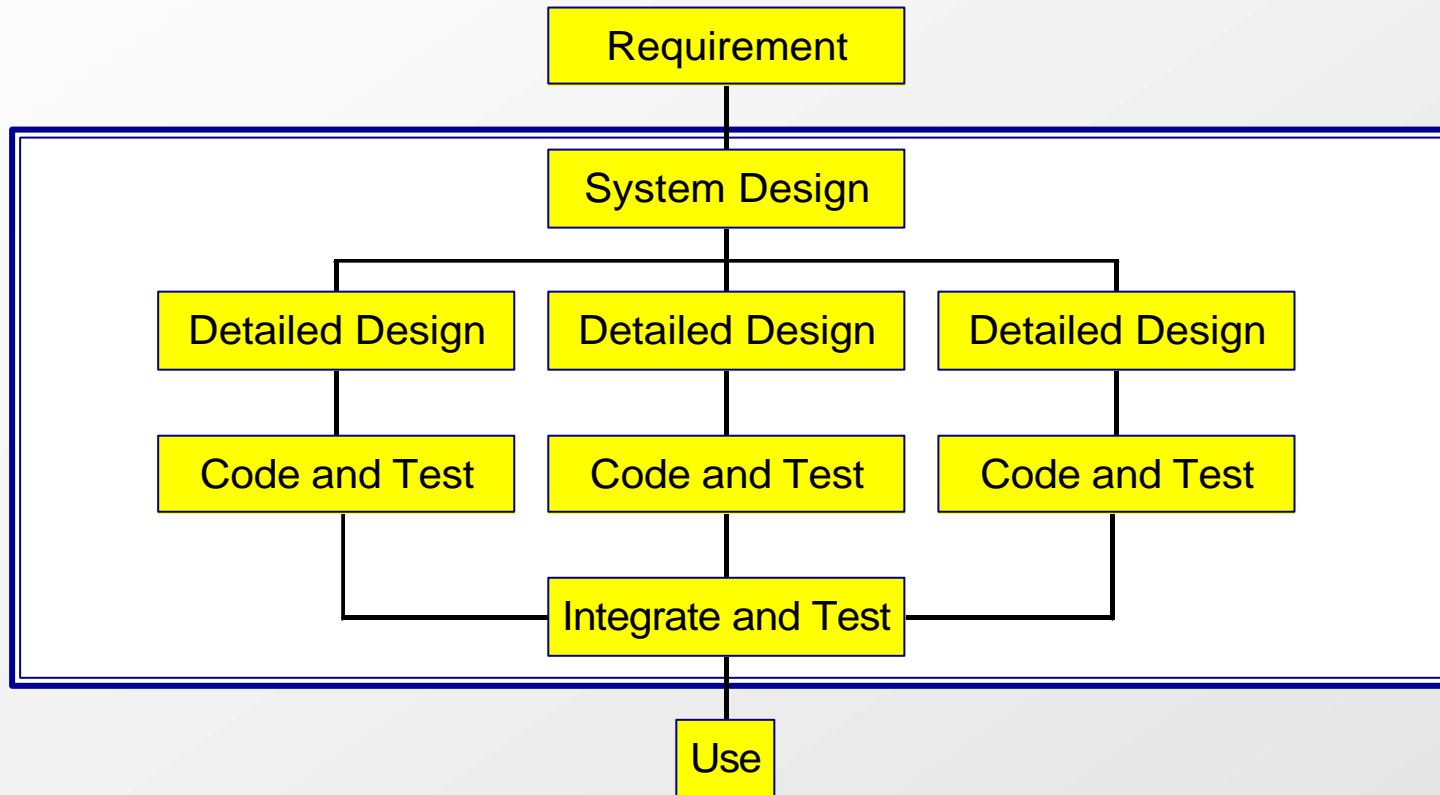




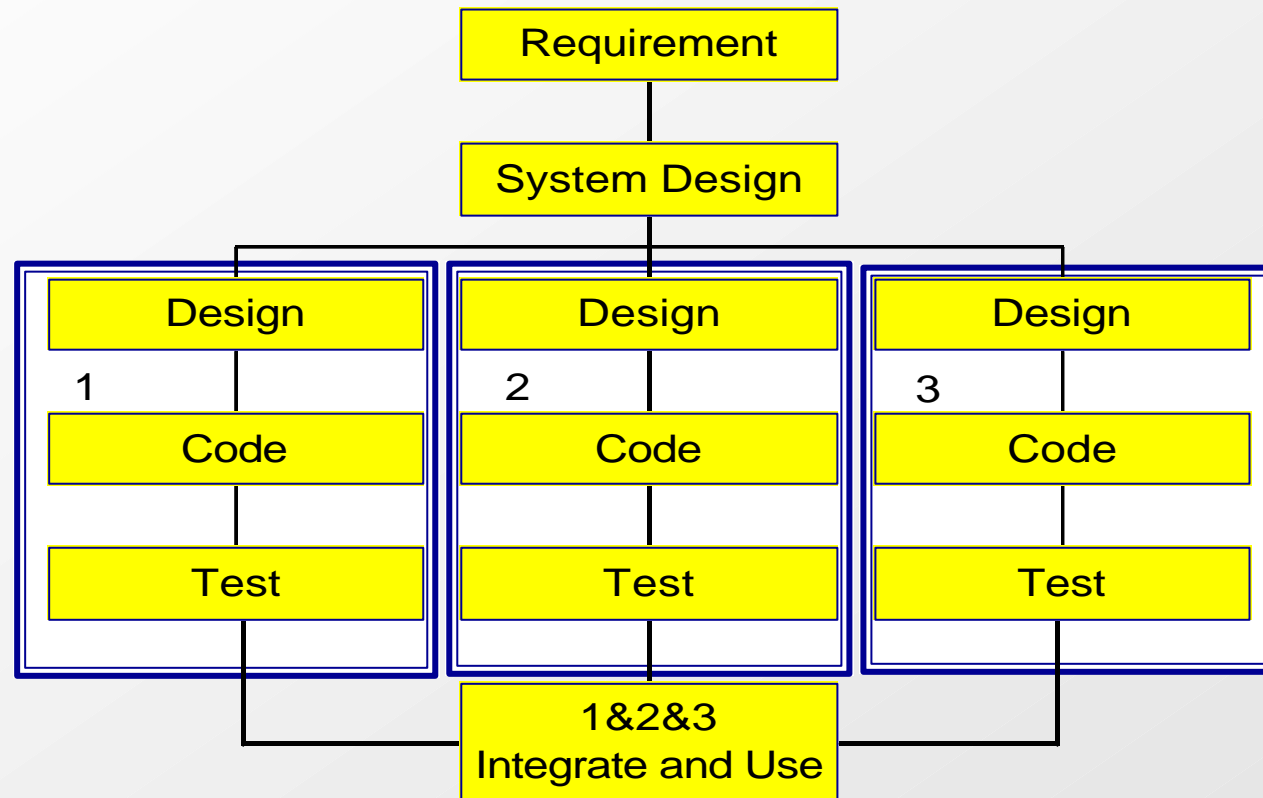
# Software development strategies

- ✍ One-shot
- ✍ Phased incremental
- ✍ Evolutionary incremental
- ✍ Prototyping
- ✍ Multi-shot

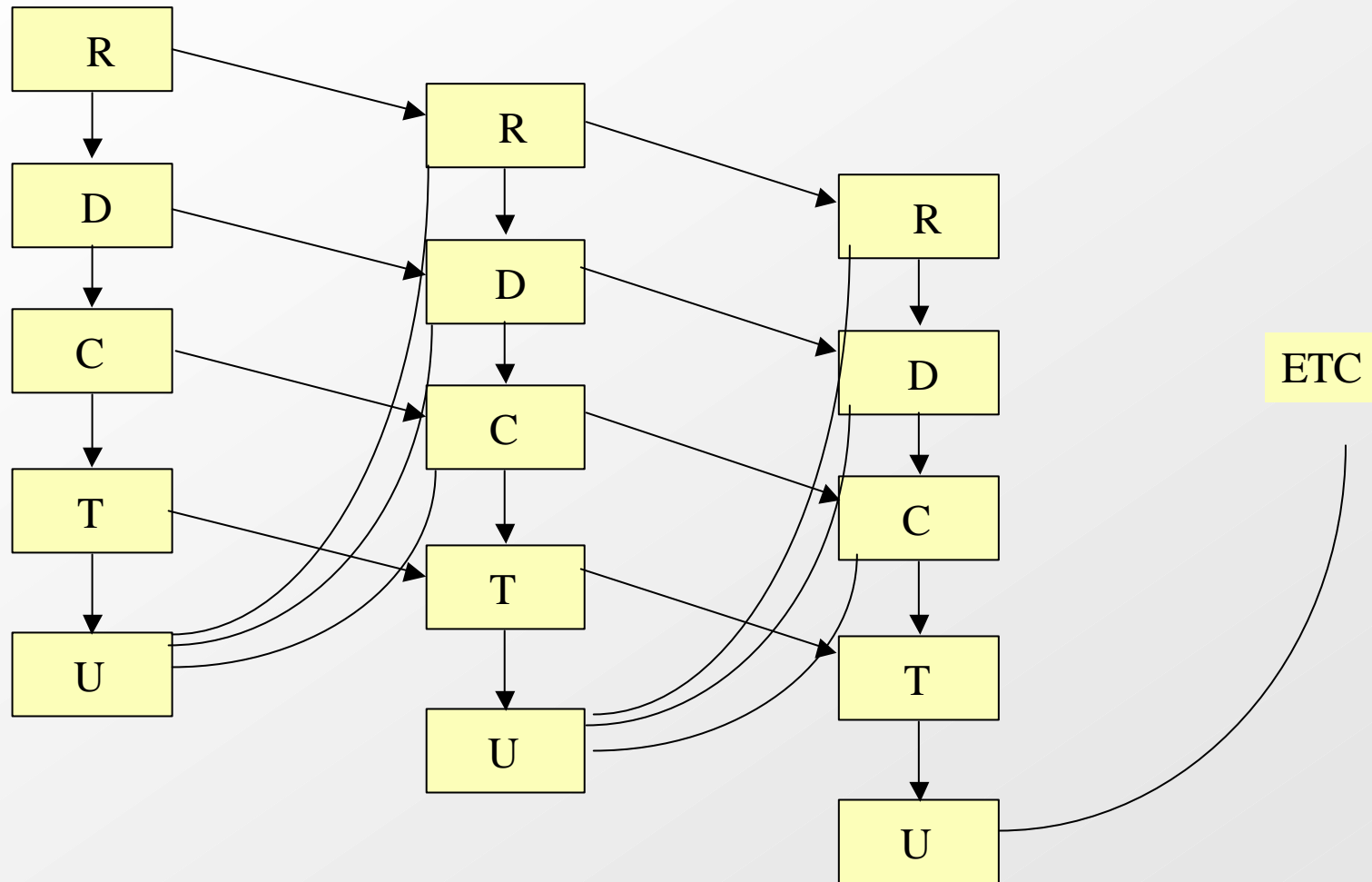
# One-shot development



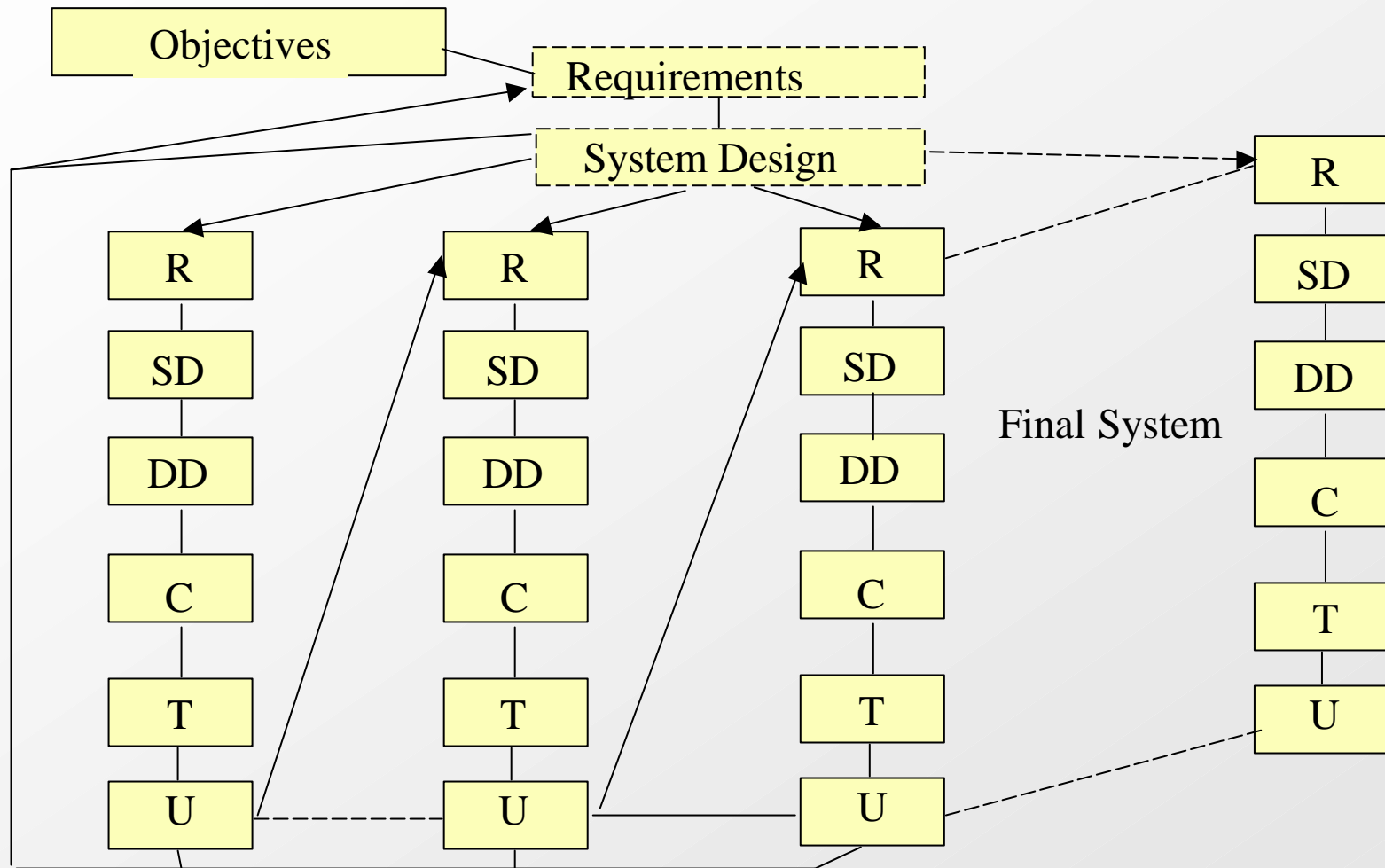
# Phased incremental development



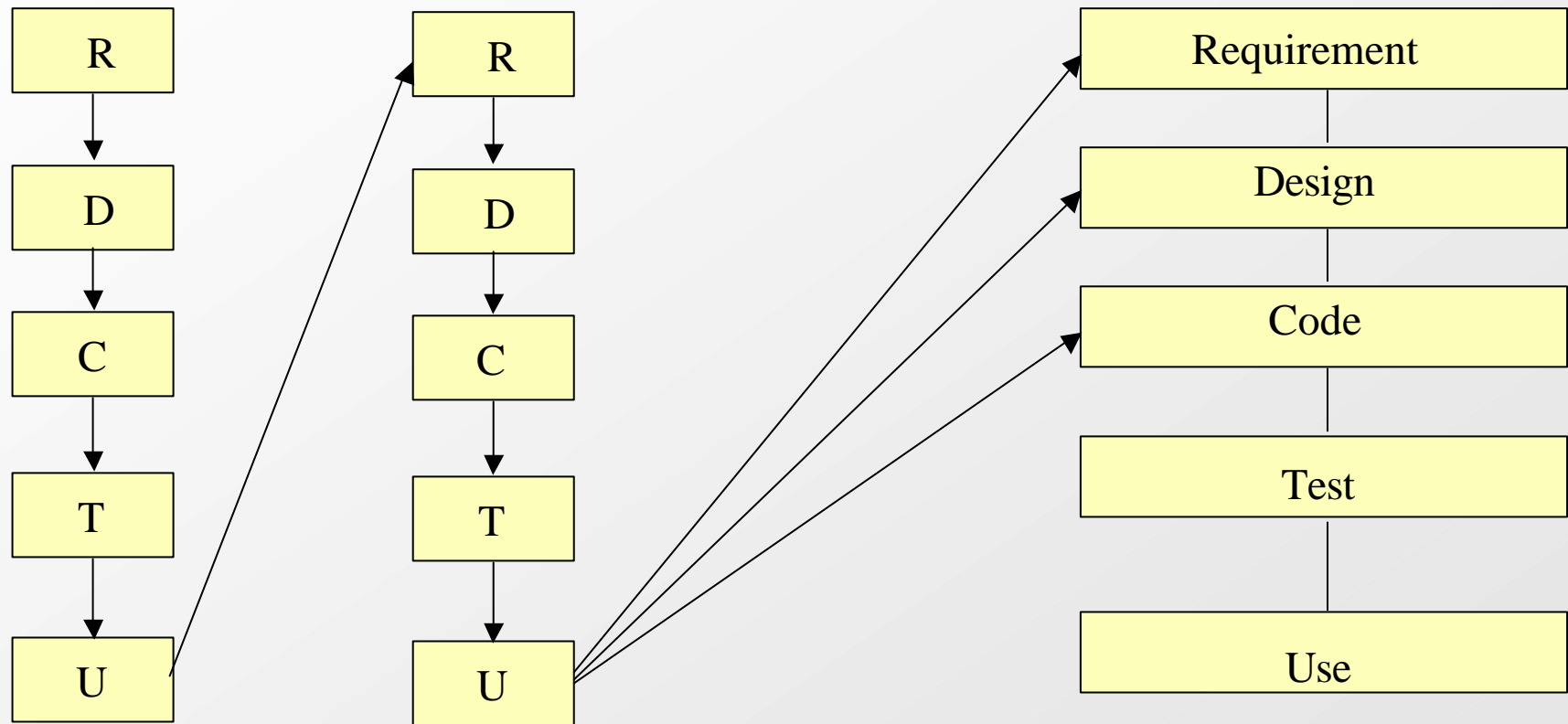
# Evolutionary development



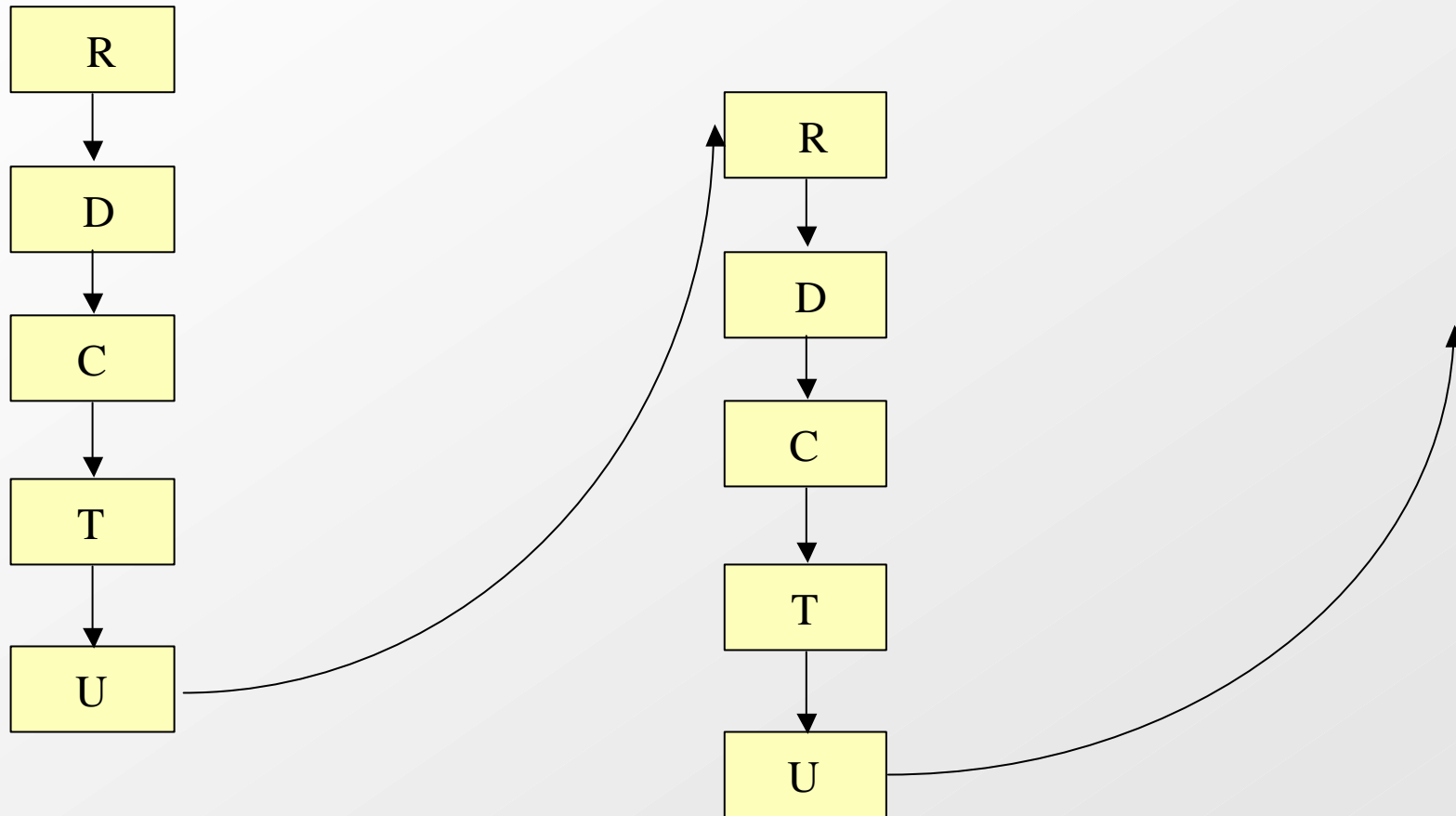
# Evolutionary incremental Development



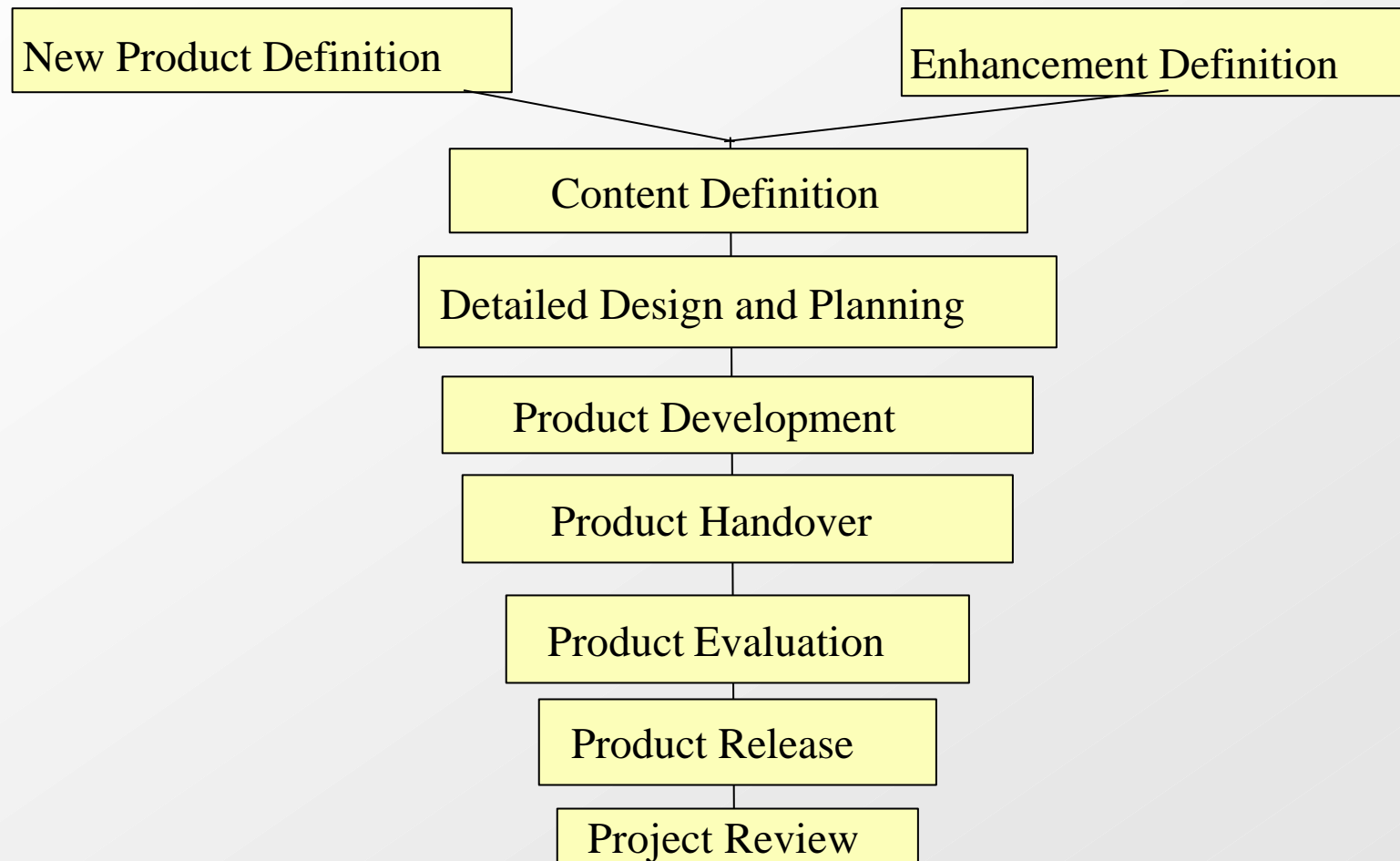
# Prototyping development



# Multi-shot development



# LifeCycle “Market Place” Products



# Strategy Summary

<b>Strategy</b>	<b>Characteristics</b>	<b>Used When</b>
<b>One-shot</b>	Linear Progression, fast if req not changed	Requirement, development and target environment well understood
<b>Phased incremental</b>	Sub-system at a time	System can be partitioned into independent modules
<b>Evolutionary Development</b>	Continuous change	Requirements subject to continual change
<b>Prototyping</b>	Working model of parts of the system	Requirement, design or environment not-well understood
<b>Multi-shot</b>	Succession of independent developments	Requirement very fuzzy system can not be planned

# Choosing a life cycle model

