Software Engineering & SDLC Models

- Sanghamitra De

Software

- Software is instructions (computer programs) that when executed provide desired function and performance.

  - ✓ Software is developed or engineered; it is not manufactured in the classical sense.
  - ✓ Software doesn't "wear out."
  - ✓ Although the industry is moving toward component-based assembly, most software continues to be custom built.

Software Engineering

- Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

FIGURE

Software engineering layers

Software Engineering: umbrella activities

- Software project tracking and control
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Document preparation and production
- Reusability management
- Measurement
- Risk management

SEI CMM

- SEI uses an assessment that results in a five point grading scheme.
- The grading scheme determines compliance with a capability maturity model (CMM) that defines key activities required at different levels of process maturity.
- The SEI approach provides a measure of the global effectiveness of a company's software engineering practices and establishes five process maturity levels

Process Maturity Levels

- Level 1: Initial
- Level 2: Repeatable
- Level 3: Defined
- Level 4: Managed
- Level 5: Optimizing

SEI has associated key process areas (KPAs) with each of the maturity levels.

Feasibility Analysis

- Simply speaking, **feasibility analysis** is simply an assessment of the practicality of a proposed plan or method.

Feasibility types

- Operational feasibility
- Technical feasibility
- Economic feasibility

Software Development Life Cycle (SDLC)

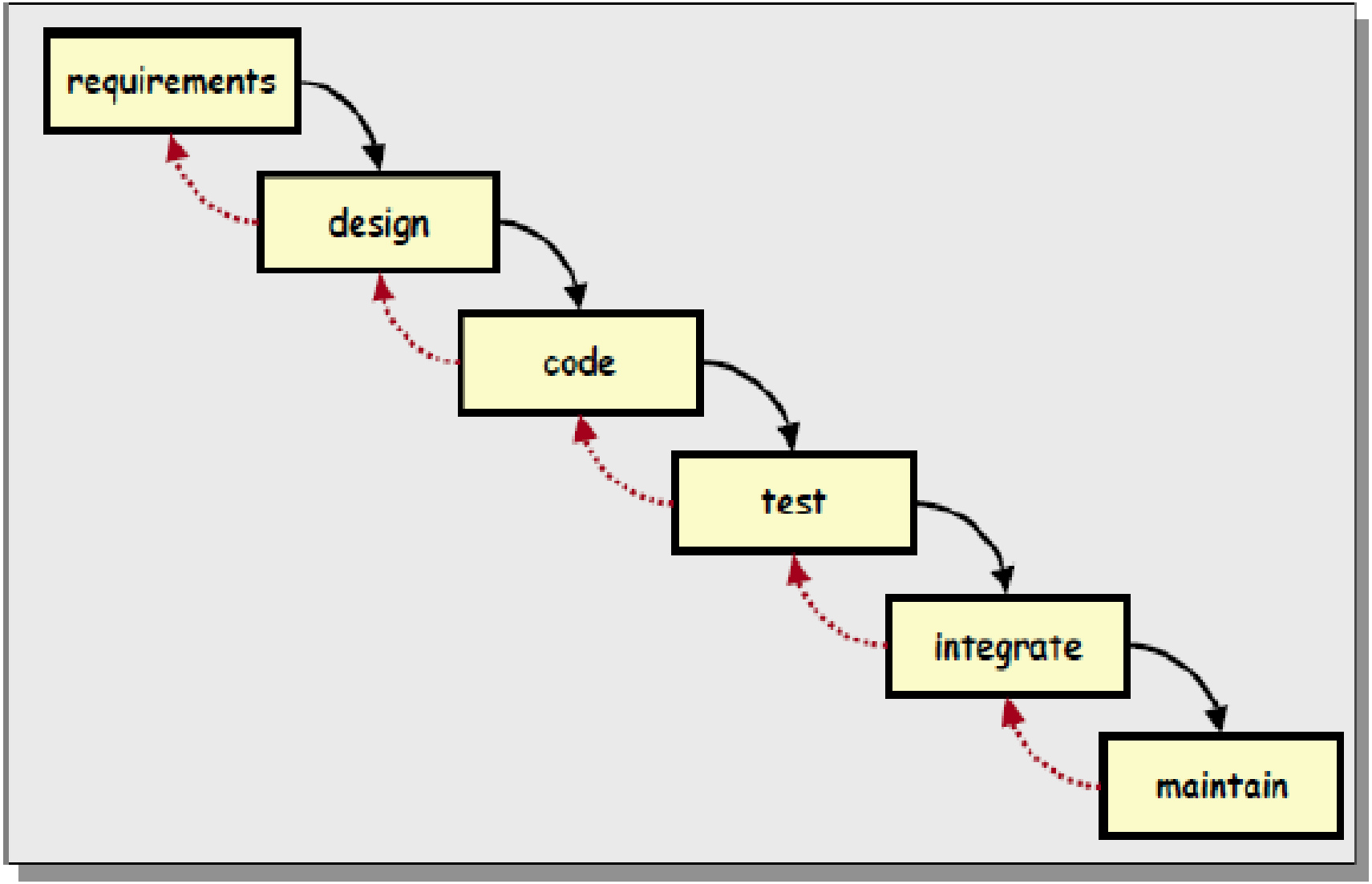
- SDLC refers to a methodology for developing systems. It provides a consistent framework of tasks and deliverables needed to develop systems.

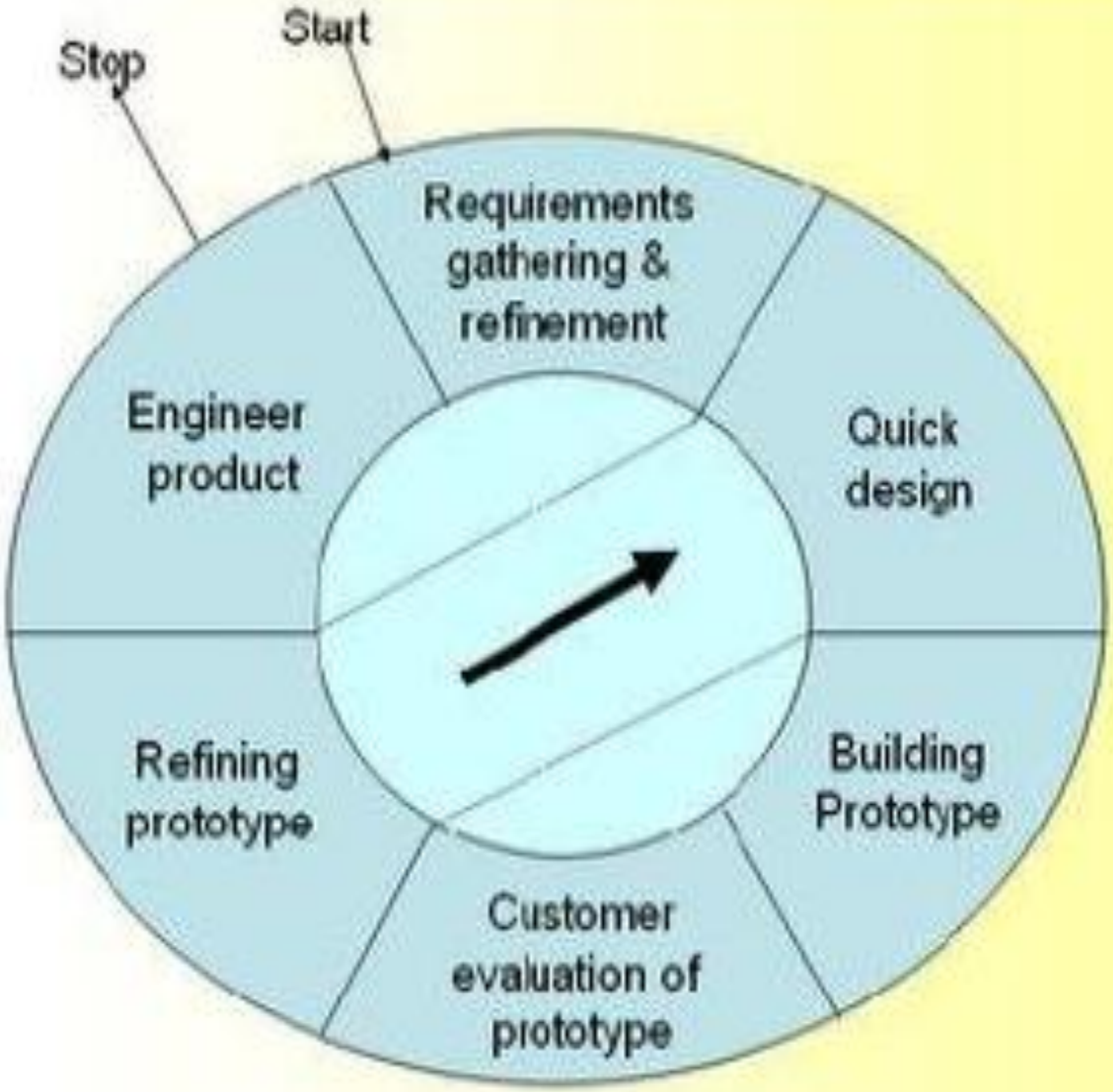
SDLC Phases

- Feasibility analysis
- Systems analysis & requirements analysis
- Systems design & Detailed design
- Implementation (Development/Coding)
- Unit & Integration testing
- Installation & deployment
- Documentation & Maintenance

Waterfall Model

Source: Adapted from Dorfman, 1997, p7
see also: van Vliet 1999, p50

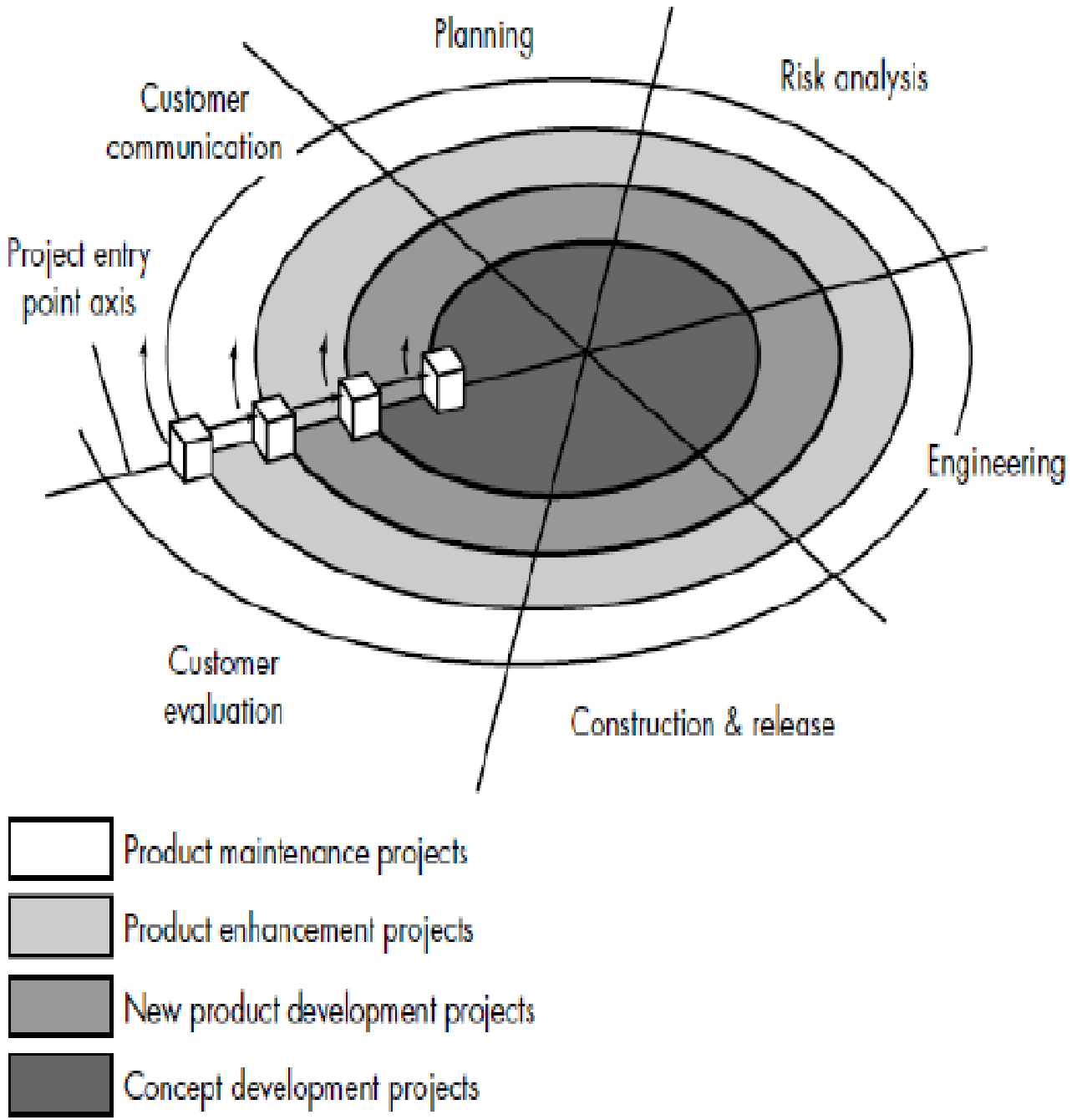


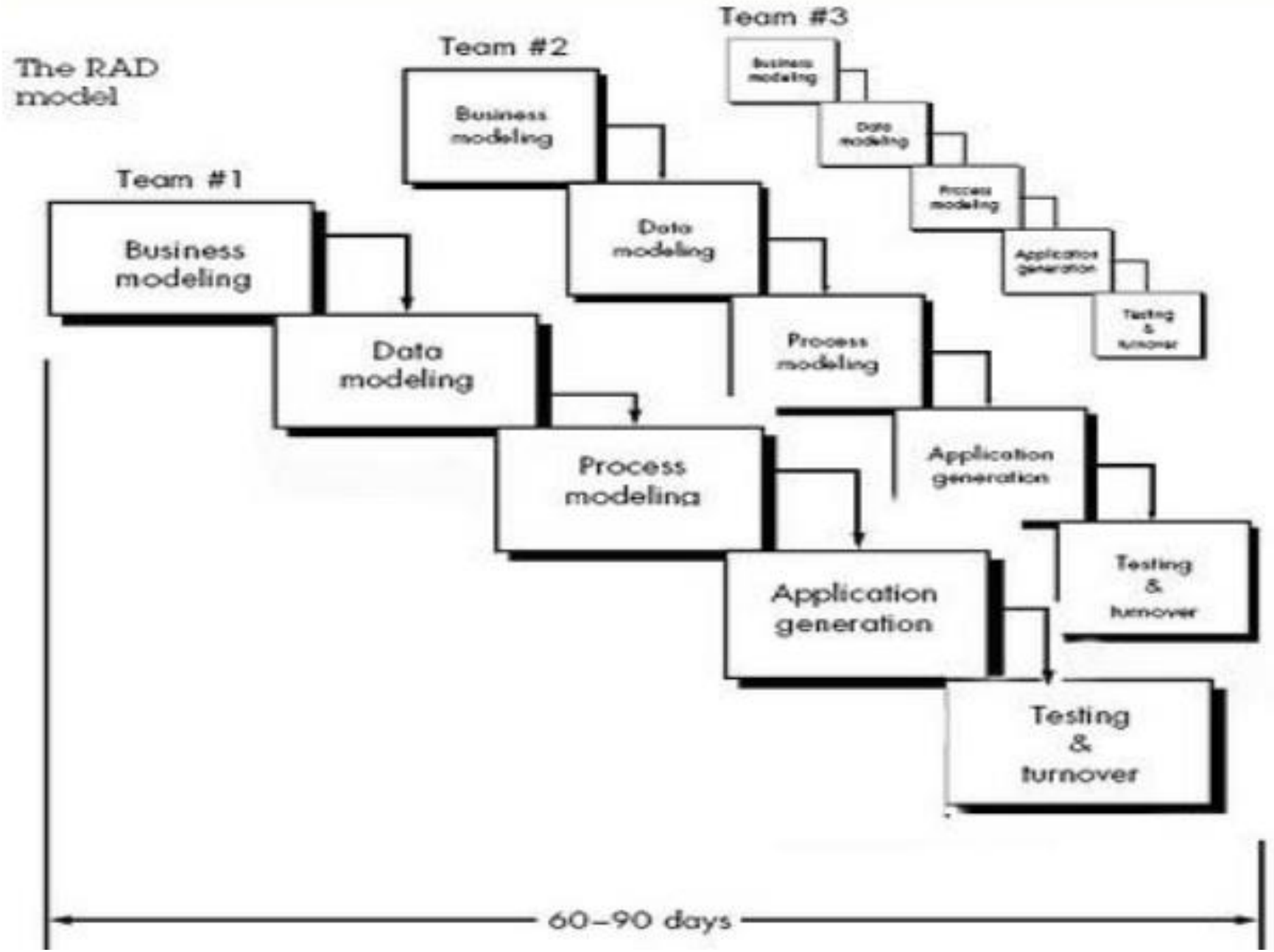


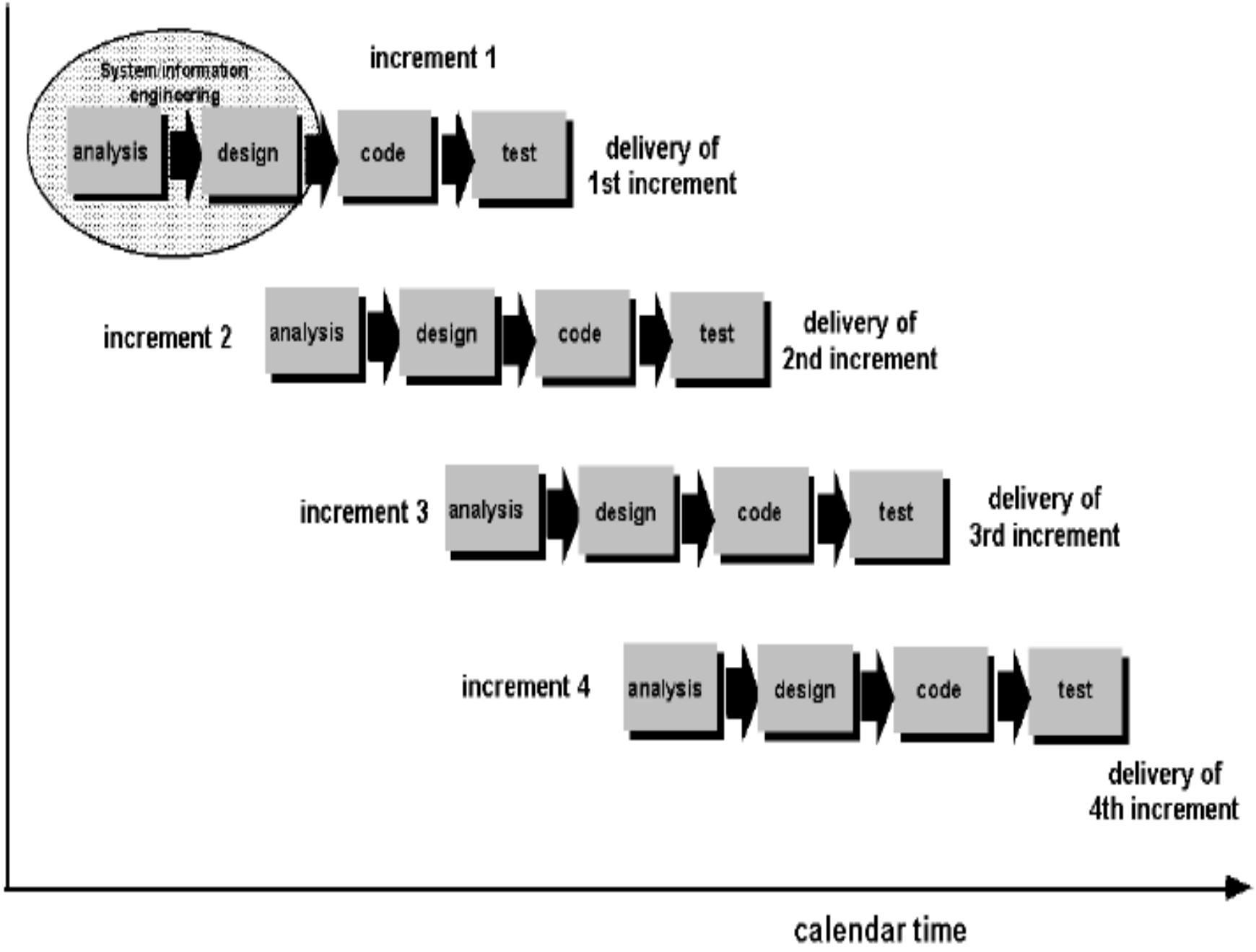
PROTOTYPE MODEL

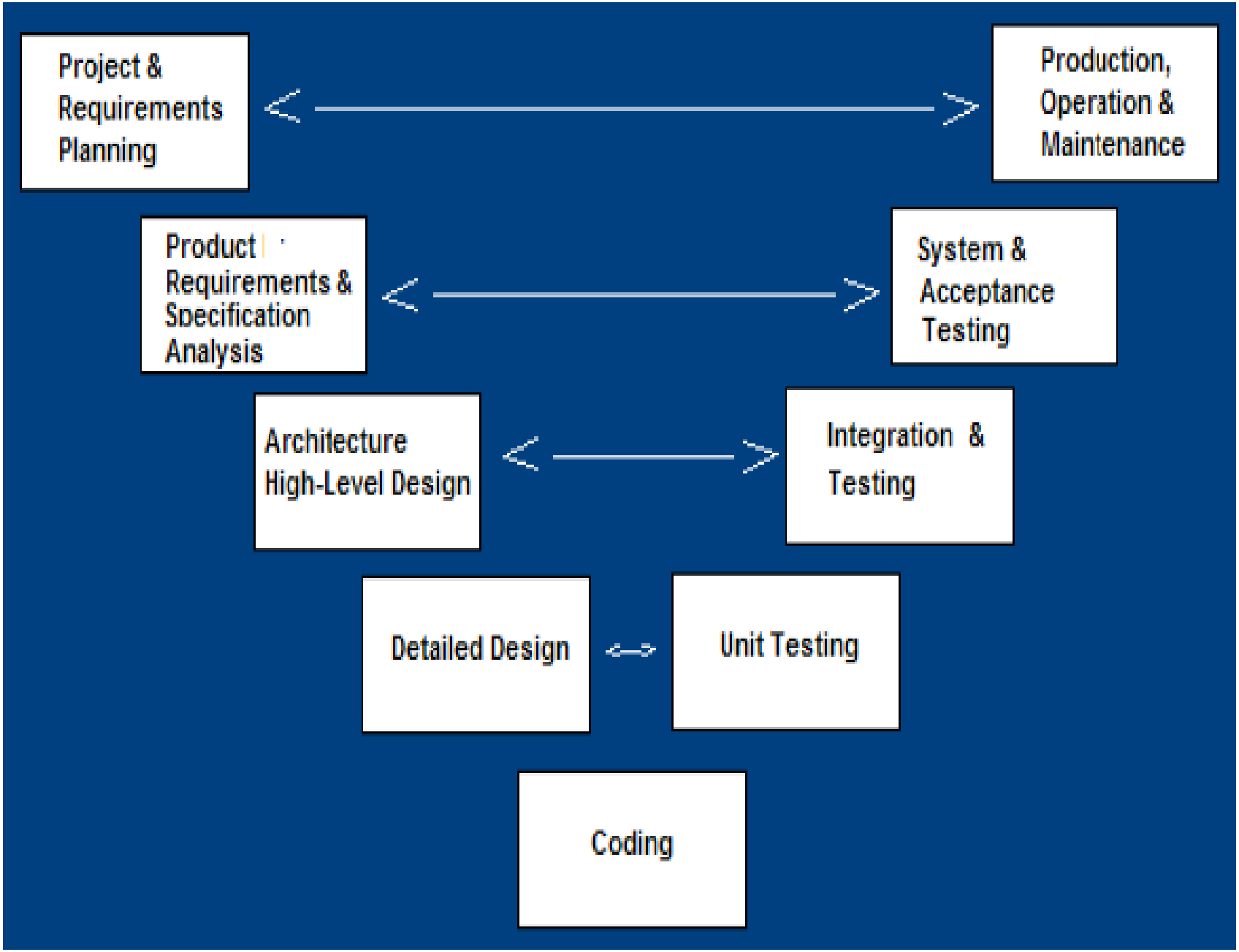
FIGURE

A typical spiral model




The RAD model
Team #1
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team #2
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team #3
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
60–90 days


System information engineering
increment 1
analysis
design
code
test
delivery of 1st increment
increment 2
analysis
design
code
test
delivery of 2nd increment
increment 3
analysis
design
code
test
delivery of 3rd increment
increment 4
analysis
design
code
test
delivery of 4th increment
calendar time


Project & Requirements Planning
Production, Operation & Maintenance
Product Requirements & Specification Analysis
System & Acceptance Testing
Architecture High-Level Design
Integration & Testing
Detailed Design
Unit Testing
Coding

Thank You…