

Unit I

Introduction

Product Life Cycles

- ▶ Products also have life cycles
- ▶ The Systems Development Life Cycle (SDLC) is a framework for describing the phases involved in developing and maintaining information systems
- ▶ Typical SDLC phases include planning, analysis, design, implementation, and support

Steps in SDLC

- ▶ Concept Exploration
- ▶ System exploration
- ▶ Requirements
- ▶ Design
- ▶ Implementation
- ▶ Installation
- ▶ Operations and support
- ▶ Maintenance
- ▶ Retirement

Process & Process Model

- ▶ **Software Process**
 - ▶ the set of activities, methods, and practices that are used in the production and evolution of software
- ▶ **Software Process Model**
 - ▶ one specific embodiment of a software process architecture

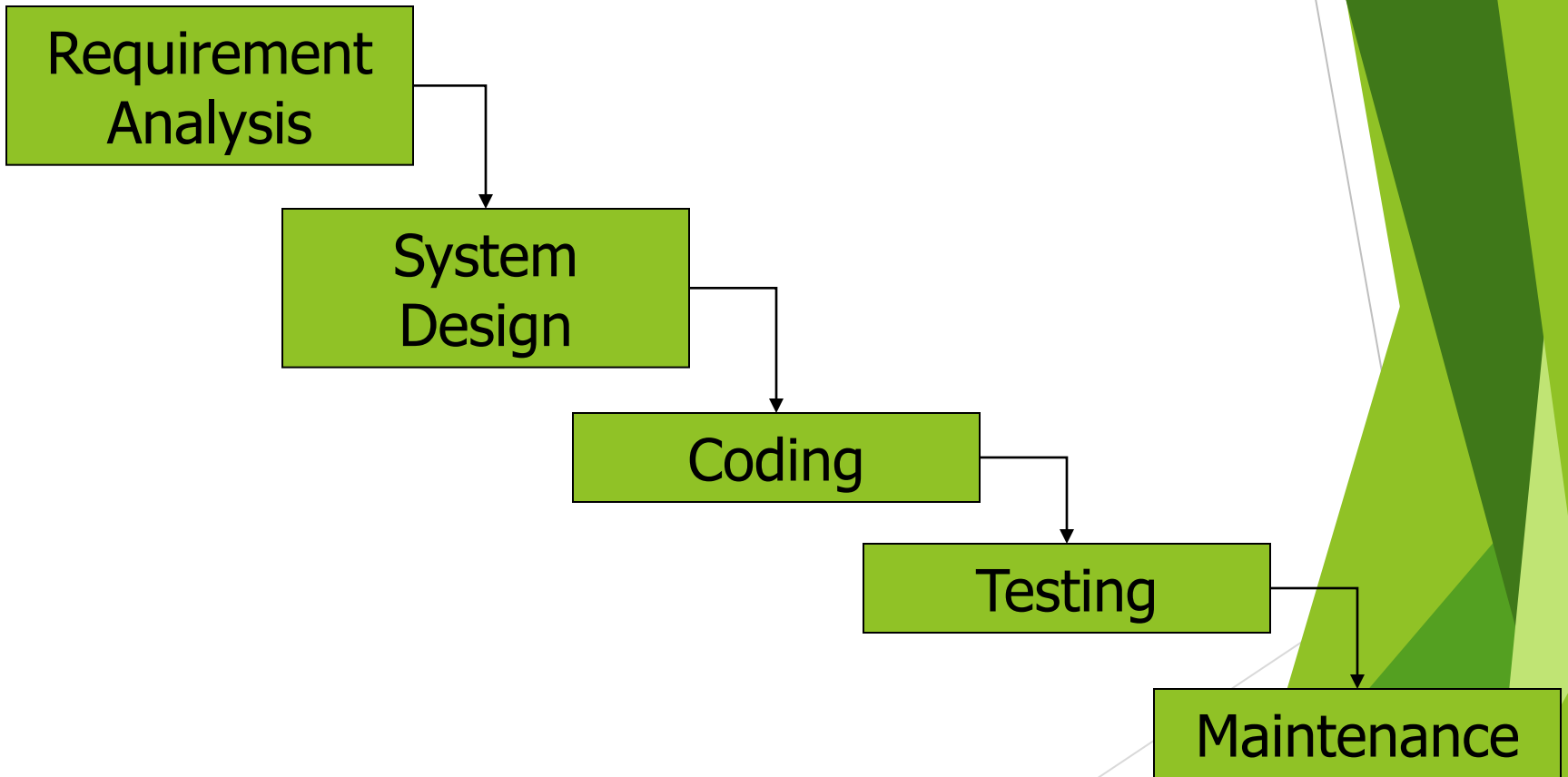
Why Modeling?

- ▶ To provide a common understanding
- ▶ To locate any inconsistencies, redundancies and omissions
- ▶ To reflect the development goals and provide early evaluation
- ▶ To assist development team to understand any special situation

Sample SDLC Models

- ▶ Waterfall model: has well-defined, linear stages of systems development and support
- ▶ Spiral model: shows that software is developed using an iterative or spiral approach rather than a linear approach
- ▶ Incremental release model: provides for progressive development of operational software
- ▶ RAD model: used to produce systems quickly without sacrificing quality
- ▶ Prototyping model: used for developing prototypes to clarify user requirements

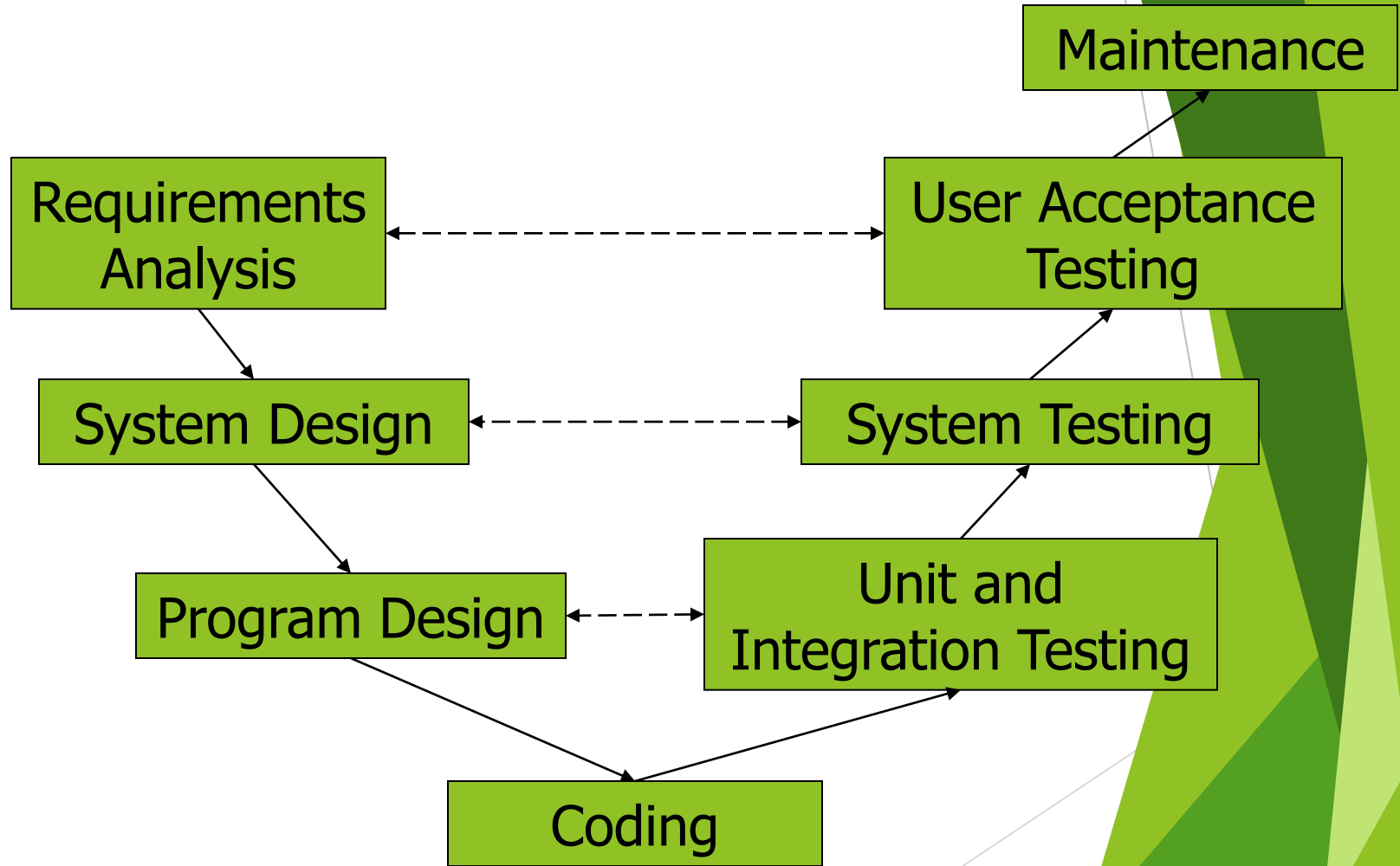
Waterfall Model



Waterfall Model (cont'd)

- ▶ classical
- ▶ one-shot approach
- ▶ effective control
- ▶ limited scope of iteration
- ▶ long cycle time
- ▶ not suitable for system of high uncertainty

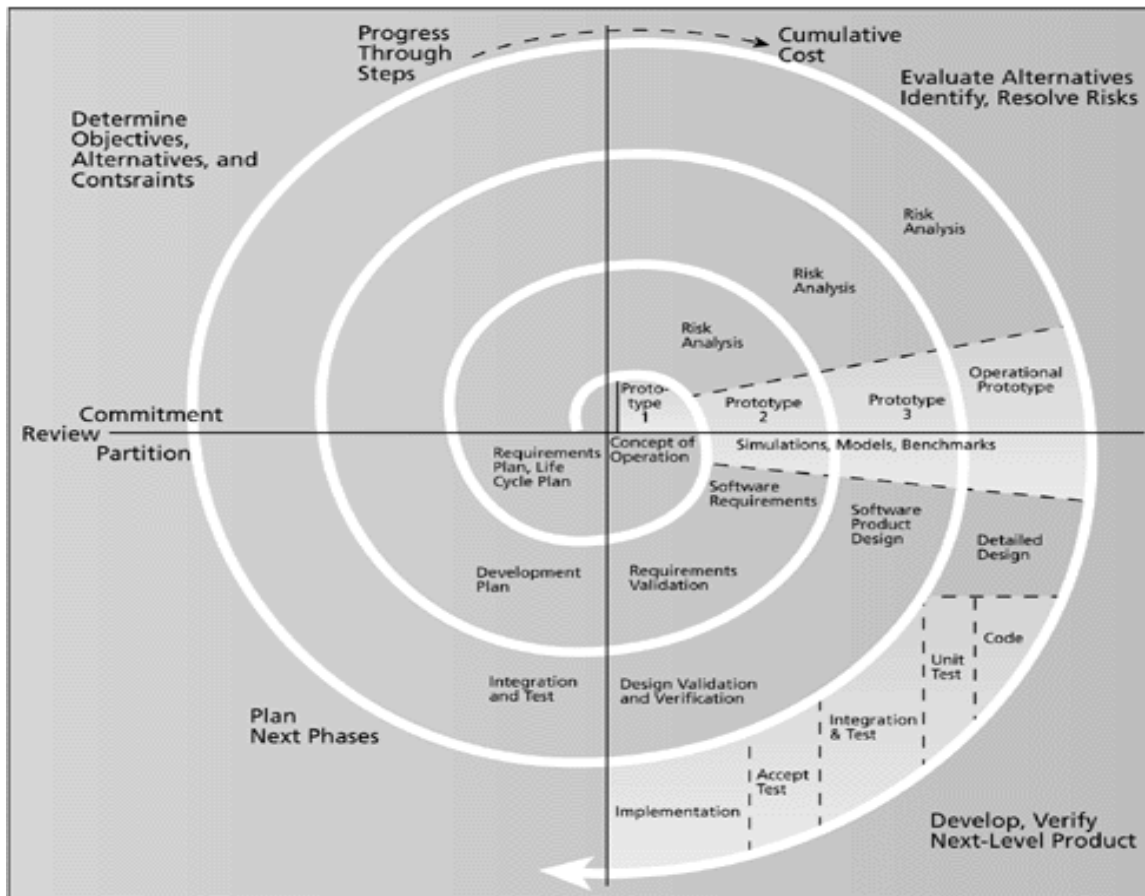
V Model



V Model (cont'd)

- ▶ Additional validation process introduced
- ▶ Relate testing to analysis and design
- ▶ Loop back in case of discrepancy

Spiral Model (adapted from Boehm 1987)



Spiral Model (cont'd)

- ▶ Evolutionary approach
- ▶ Iterative development combined with risk management
- ▶ Risk analysis results in “go, no-go” decision

Spiral Model (cont'd)

- ▶ Four major activities
 - ▶ Planning
 - ▶ Risk analysis
 - ▶ Engineering
 - ▶ Customer evaluation

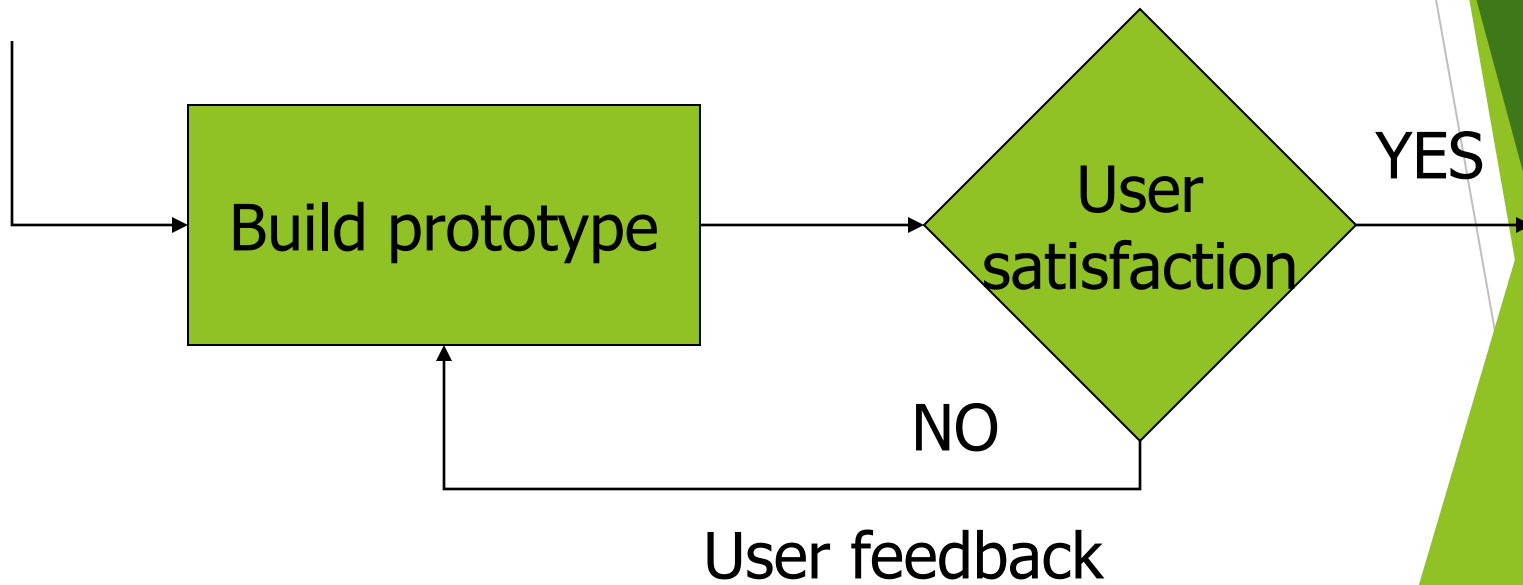
Prototyping Model

- ▶ Goals
 - ▶ meet users' requirements in early stage
 - ▶ reduce risk and uncertainty

Classification of Prototype

- ▶ **Throw-away**
 - ▶ After users agree the requirements of the system, the prototype will be discarded.
- ▶ **Evolutionary**
 - ▶ Modifications are based on the existing prototype.
- ▶ **Incremental**
 - ▶ Functions will be arranged and built accordingly.

Prototyping Model



Benefits of Prototyping

- ▶ Learning by doing
- ▶ Improved communication
- ▶ Improved user involvement
- ▶ Clarification of partially-known requirements

Prototyping Sequences

- ▶ Requirements gathering
- ▶ Quick design
- ▶ Prototype construction
- ▶ Customer evaluation
- ▶ Refinement
- ▶ Loop back to quick design for fine tuning
- ▶ Product engineering

Benefits of Prototyping

- ▶ Demonstration of the consistency and completeness of a specification
- ▶ Reduced need for documentation
- ▶ Reduced maintenance costs
- ▶ Feature constraint
- ▶ Production of expected results

Drawbacks of Prototyping

- ▶ Users sometimes misunderstand the role of the prototype
- ▶ Lack of project standards possible
- ▶ Lack of control
- ▶ Additional expense
- ▶ Close proximity of developers

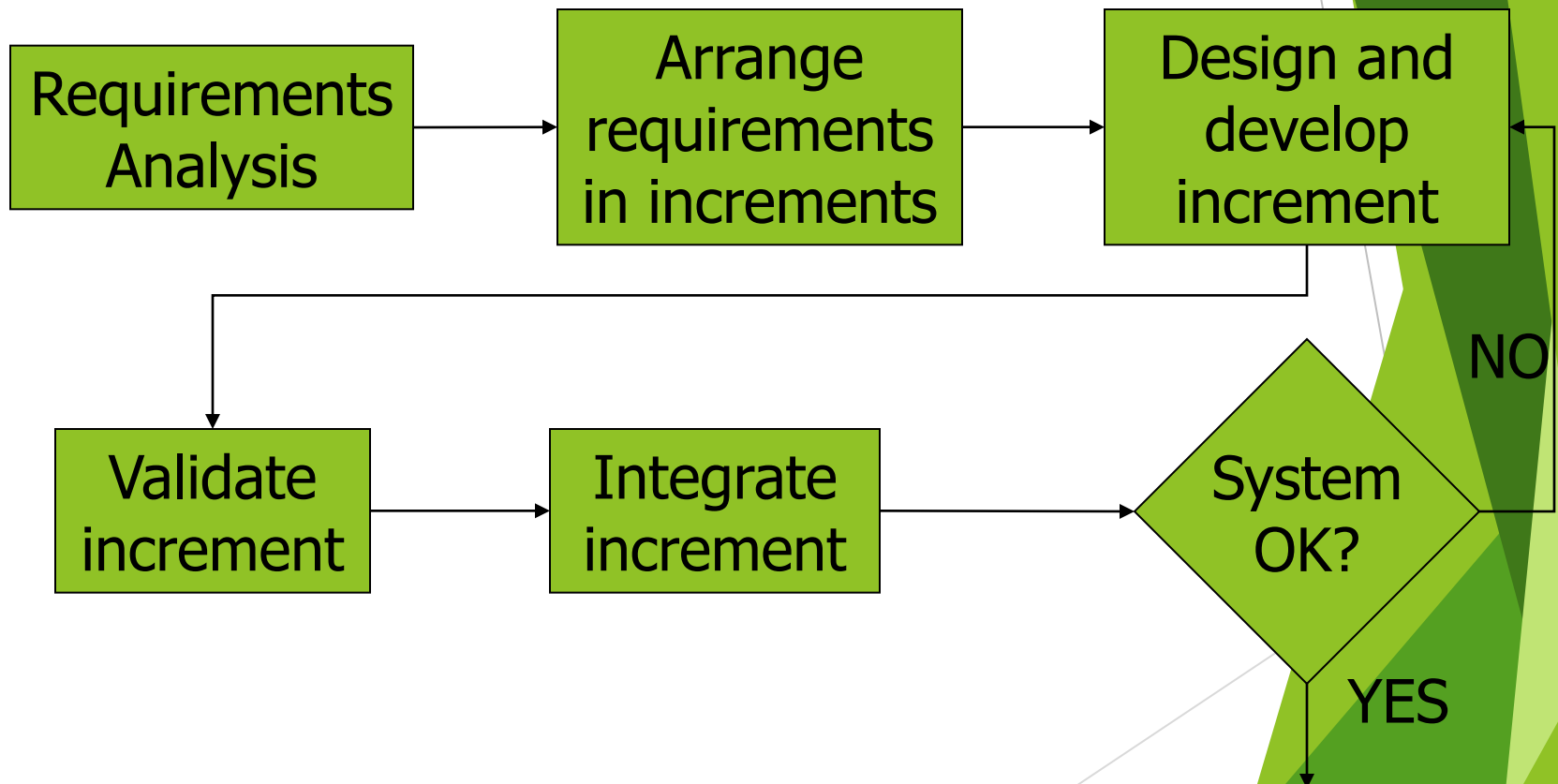
Forms of Prototypes

- ▶ Mock-ups
- ▶ Simulated interaction
- ▶ Partial working model

Incremental Model

- ▶ Break system into small components
- ▶ Implement and deliver small components in sequence
- ▶ Every delivered component provides extra functionality to user

Incremental Model (cont'd)



Iterative Model

- ▶ Deliver full system in the beginning
- ▶ Enhance functionality in new releases

Iterative Model (cont'd)

