

# CISC/CMPE 327 Software Quality Assurance

Queen's University, 2019-fall

Lecture #3-1  
Iterative Model  
Software Process Models - 2

# The Iterative Development Model

- Subset Development
  - The Iterative Development Process (IDP) is based on subsets
  - Begin with a subset of the requirements and develop a subset of the software product
  - The subset should:
    - satisfy immediate needs of users
    - serve as a vehicle of training for customers, and learning for developers

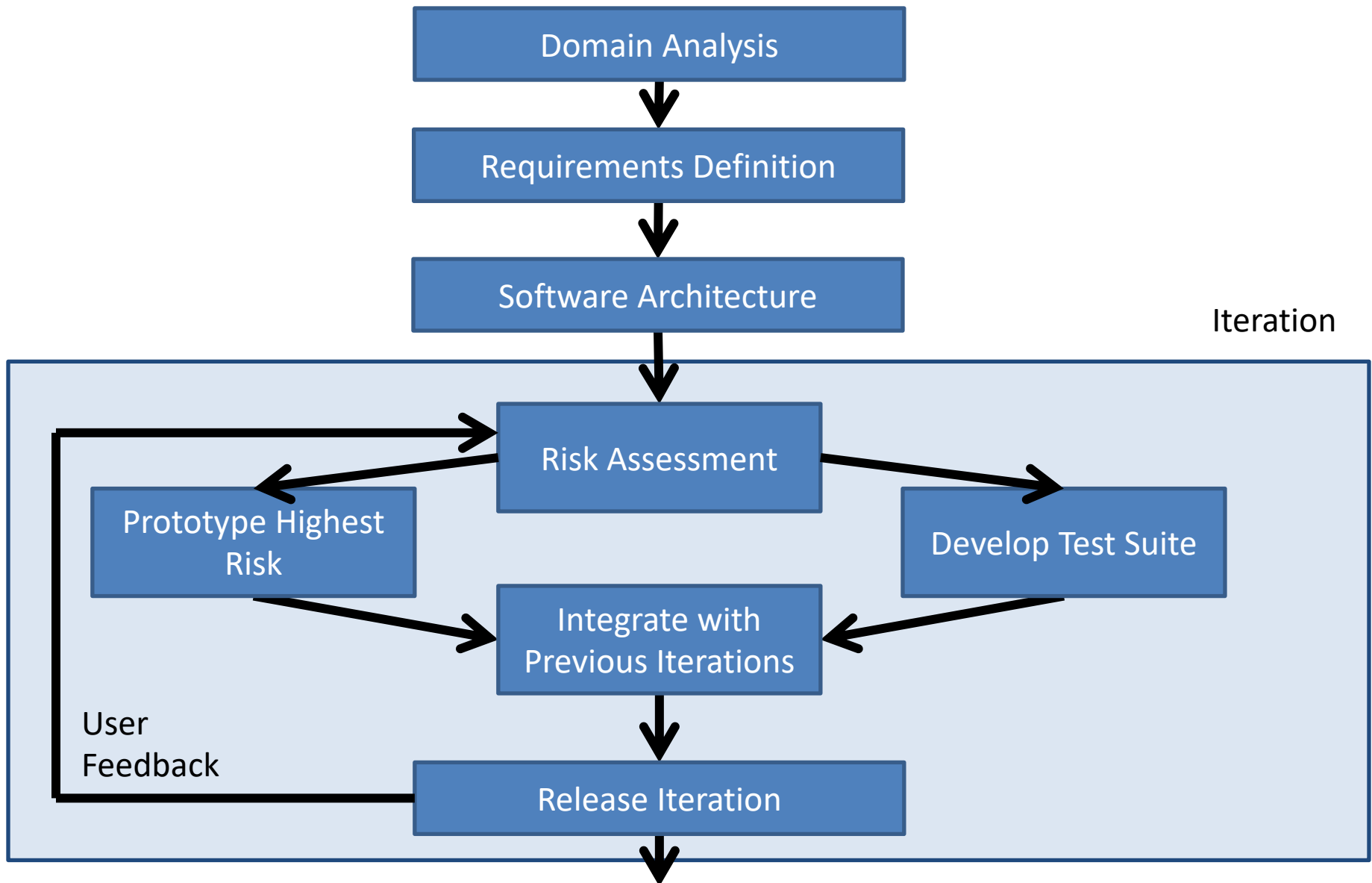
# The Iterative Development Model

- Sequence of Intermediate Products
  - Analysis of the subset product leads to modifications to the design and requirements, from which we build a new (hopefully larger) subset product
  - Design and requirements refined over a series of iterations to provide a system that meets evolving customer needs with improved design based on feedback and learning

# The Iterative Development Process

- Iterative Development Process
  - Analysis of the problem domain and definition of requirements begins process as usual
  - Need initial architecture design to begin
  - Add most critical remaining features each cycle
  - Quality control, development of test suite for new features on each iteration

# The Iterative Development Process



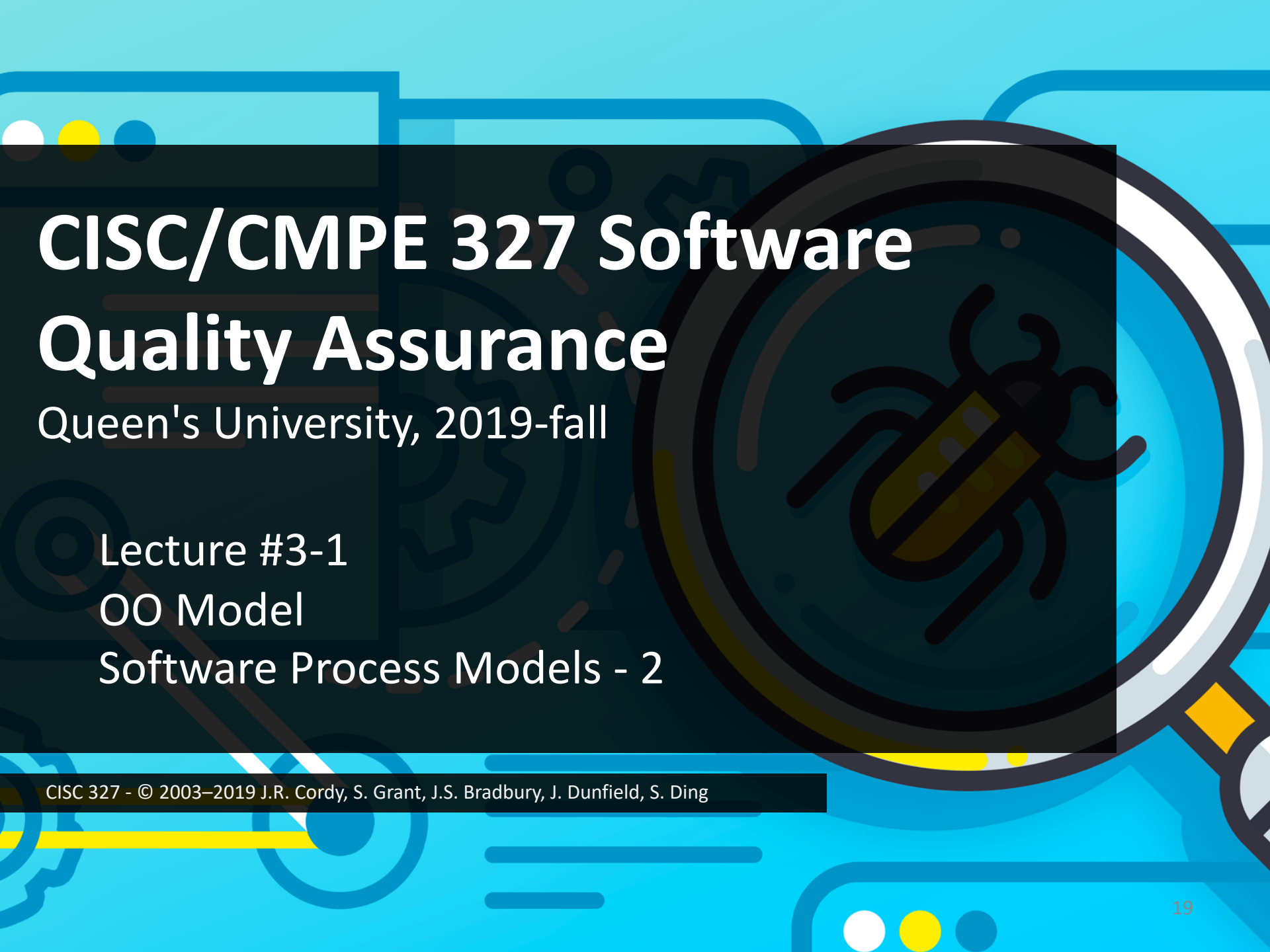
# Drawbacks of the Iterative Process

- Needs Small Team

- Process does not allow for large scale parallel development, depends on focussing on **one** remaining risk at a time
- Works best with relatively **small** teams

- Needs Early Architecture

- Requires early design of overall **architecture**, difficult to change later
- But when architecture can be settled early, has been very successful at producing significant, very **high quality** products, e.g., IBM's **OS/2** system



# CISC/CMPE 327 Software Quality Assurance

Queen's University, 2019-fall

Lecture #3-1

OO Model

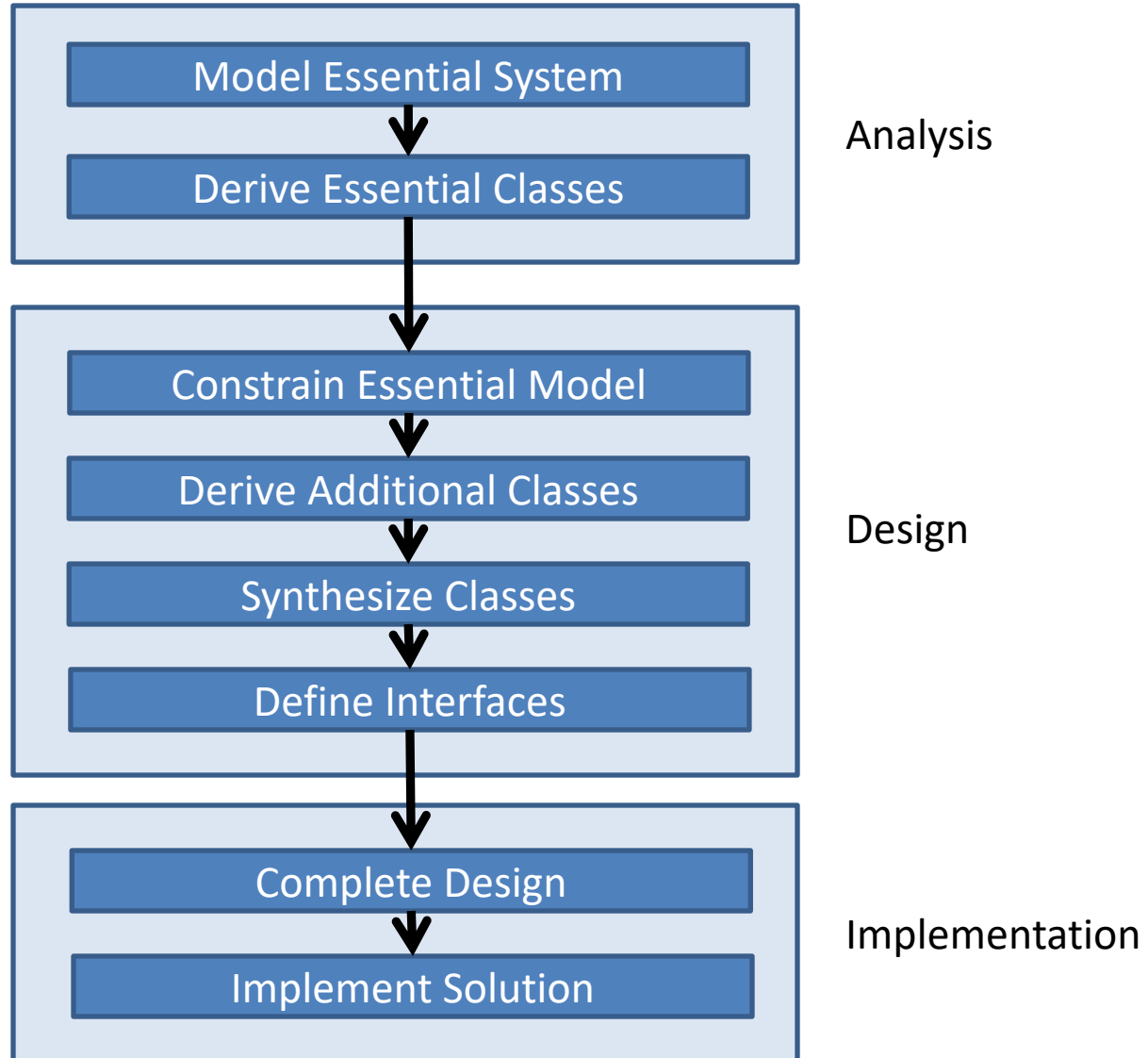
Software Process Models - 2

# Object-Oriented Development

- OO Process Models
  - Many OO process models are proposed, based on OOAD ("Object-Oriented Analysis and Design")
  - All have three major phases:
    - Analysis: model the "essential system" to represent user requirements, and design implementation-independent "essential classes"
    - Design: constrain and refine essential classes to be implemented on particular implementation environment, derive additional classes
    - Implementation: define class interfaces and implementation methods, then code and unit test all classes



# Object-Oriented Development



# OO Development Process - Analysis

- 1) Model the Essential System
  - Create a "user view" of the system
  - Model essential activities, essential solution data, and how they are related
  - Quality control: requirements reviews (inspection)

# OO Development Process - Analysis

- 2) Derive Essential Classes
  - "Carve" out candidate **essential classes** from the essential model using data-flow diagrams, process, and data specifications
  - Quality control: **design reviews** (inspection)

# OO Development Process - Design

- 3) Constrain the Essential Model
  - Modify essential model to fit within constraints of target implementation environment
  - Map essential activities and data to implementation processors (hardware/software) and containers (memory/files)

# OO Development Process - Design

- 4) Derive Additional Classes
  - Additional **classes** and **methods** specific to implementation environment added to support additional activities added while constraining the essential model

# OO Development Process - Design

- 5) Synthesize Classes
  - Essential classes and additional classes **refined** and **organized** into a class hierarchy
  - Final classes chosen to maximize **reuse**
  - Quality control: **design review** (inspection)
- 6) Define Interfaces
  - Class definitions written for final classes

# OO Development - Implementation

- 7) Complete Design

- Design of "implementation module" completed
- Implementation module specifies methods such that each provides a single cohesive function
- Quality control: design review

- 8) Implement Solution

- Implementation of classes and methods is coded and validated
- Quality control: unit testing (class-wise)

# Drawbacks of the OO Process

- **Delayed Testing**
  - Development process missing intermediate results
  - Most testing **delayed** to final implementation stage
- **Architectural Inflexibility**
  - Process assumes that overall architecture can be designed in the requirements phase
  - Allows little **architectural flexibility** in design and implementation steps
  - Can lead to spaghetti results



# Summary

- Software Process
  - Spiral Model organizes and generalizes the waterfall model
  - Iterative Development Process is based on product subsets
  - Object-Oriented Development is (was?) a currently popular model with drawbacks

# Review Questions

- Waterfall model
  - Drawbacks?
  - When would it be appropriate?
- Prototyping model
  - Drawbacks?
  - Advantages?

# Review Questions

- Spiral model
  - Strengths?
  - Drawbacks?
- Iterative development
  - Strengths?
  - Drawbacks?
- Object-oriented development
  - Strengths?
  - Drawbacks?

# Summary

- Today's References
  - Sommerville, *Software Engineering*, ch. 2
  - Kan, *Metrics and Models in Software Quality Engineering*, ch. 2
- Next Time
  - Quality standards and assessment of software processes
- Then...
  - The eXtreme Programming software process