1

Kendall & Kendall
Systems Analysis and Design

Systems, Roles, and Development Methodologies

The Agile Approach

- Based on:
  - Values
  - Principles
  - Core practices

Agile Values

- Communication
- Simplicity
- Feedback
- Courage

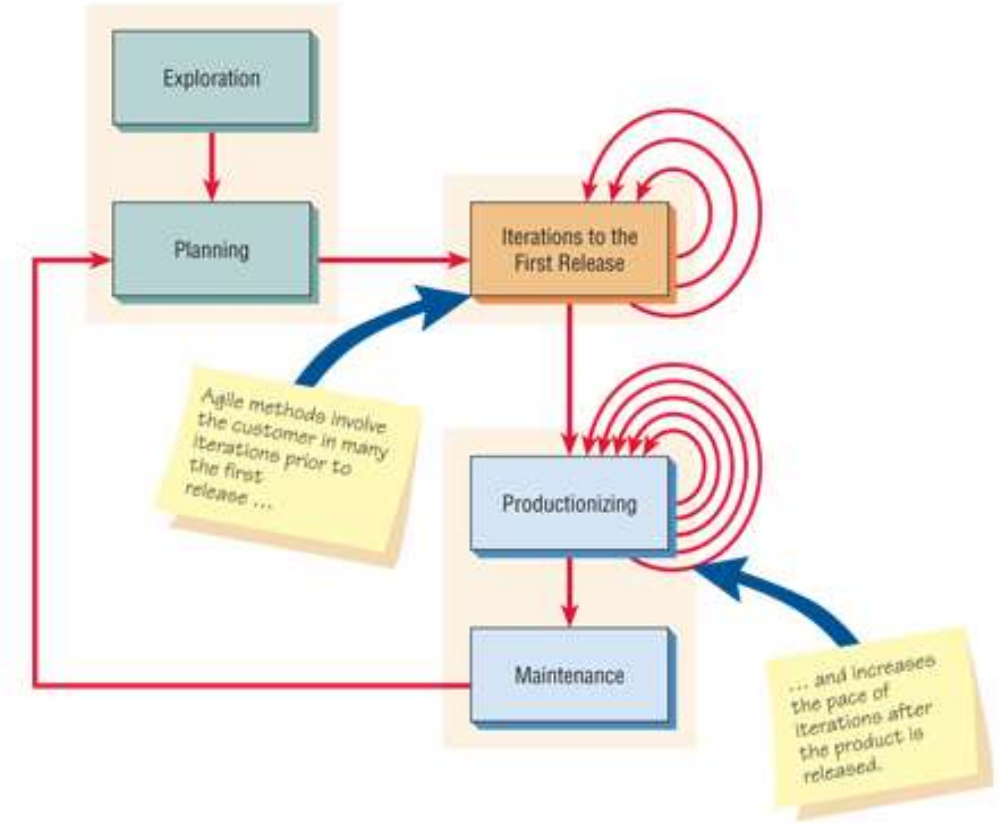
Four Agile Resources

- Resources are adjusted to ensure successful project completion
  - Time
  - Cost
  - Quality
  - Scope

Five Stages of Agile Development

- Exploration
- Planning
- Iterations to the first release
- Productionizing
- Maintenance

Agile Project Development Process (Figure 1.5)



Object-Oriented (O-O) Systems Analysis and Design

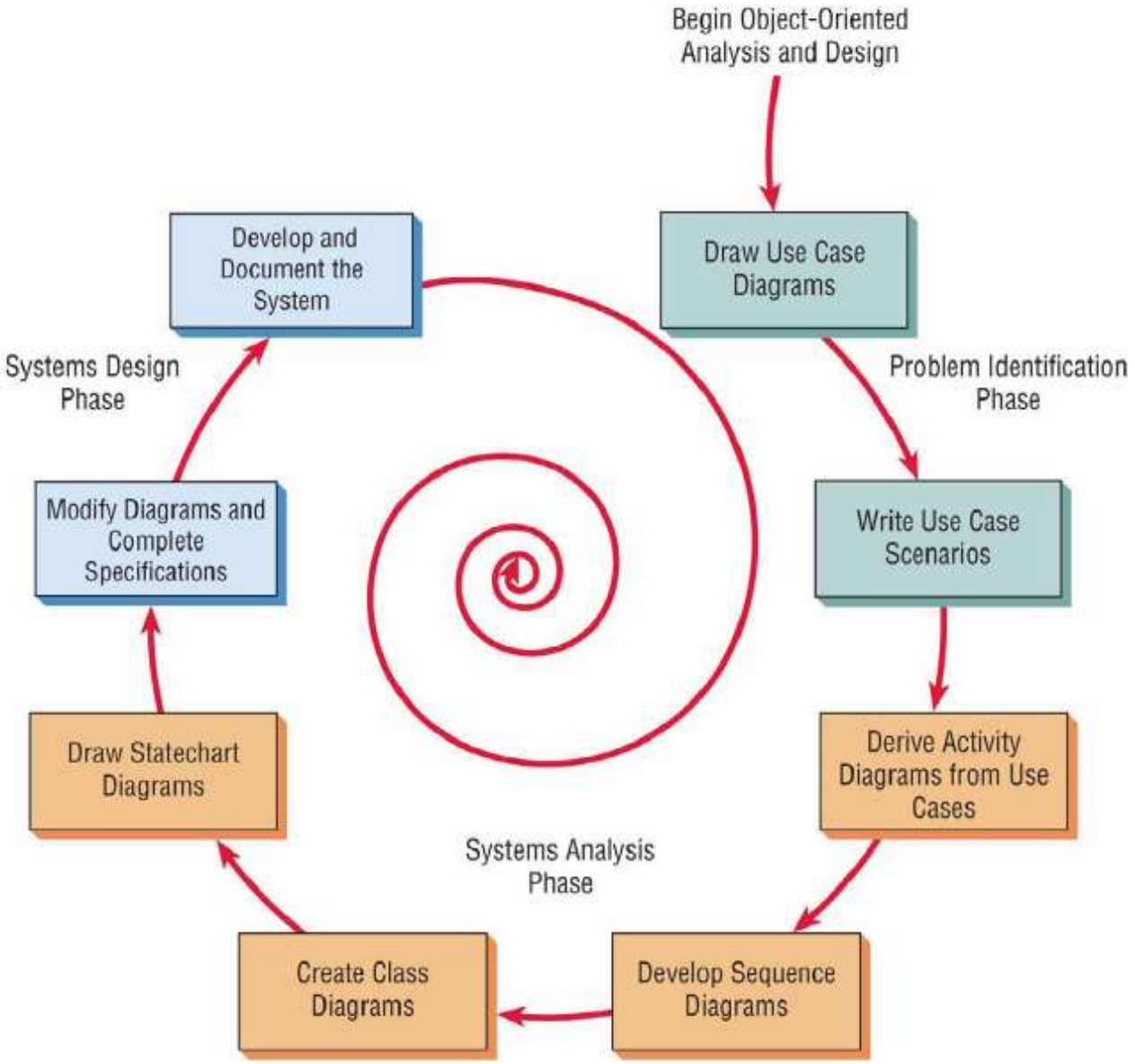
- Alternate approach to the structured approach of the SDLC that is intended to facilitate the development of systems that change rapidly in response to dynamic business environments
- Analysis is performed on a small part of the system followed by design and implementation

Object-Oriented (O-O) Systems Analysis and Design

- The cycle repeats with analysis, design, and implementation of the next part and this repeats until the project is complete
- Examines the objects of a system

Unified Modeling Language (UML) Phases

- Define the use case model:
  - Use case diagram
  - Use case scenarios
- Create UML diagrams
- Develop class diagrams
- Draw statechart diagrams
- Modify the UML diagrams
- Develop and document the system


Begin Object-Oriented Analysis and Design
Draw Use Case Diagrams
Problem Identification Phase
Write Use Case Scenarios
Derive Activity Diagrams from Use Cases
Develop Sequence Diagrams
Create Class Diagrams
Draw Statechart Diagrams
Modify Diagrams and Complete Specifications
Systems Design Phase
Develop and Document the System
Systems Analysis Phase

Choosing a Method

- Choose either:
  - SDLC
  - Agile
  - Object-oriented methodologies

When to Use SDLC

- Systems have been developed and documented using SLDC
- It is important to document each step
- Upper level management feels more comfortable or safe using SDLC
- There are adequate resources and time to complete the full SDLC
- Communication of how new systems work is important

When to Use Agile

- There is a project champion of agile methods in the organization
- Applications need to be developed quickly in response to a dynamic environment
- A rescue takes place (the system failed and there is no time to figure out what went wrong)
- The customer is satisfied with incremental improvements
- Executives and analysts agree with the principles of agile methodologies

When to Use Object-Oriented

- The problems modeled lend themselves to classes
- An organization supports the UML learning
- Systems can be added gradually, one subsystem at a time
- Reuse of previously written software is a possibility
- It is acceptable to tackle the difficult problems first

Open Source Software

- An alternative of traditional software development where proprietary code is hidden from the users
- Open source software is free to distribute, share, and modify
- Characterized as a philosophy rather than simply the process of creating new software
- Examples: Linux Operating System, Apache Web Server, Mozilla Firefox

Four Types of Open Source Communities:

- Ad hoc
- Standardized
- Organized
- Commercial

Six Key Dimensions that Differentiate Open Source Communities

- General structure
- Environment
- Goals
- Methods
- User community
- Licensing

Reasons for Participating in Open Source Communities

- Rapidity with which new software can be developed and tested
- Faster to have a committed group of experts develop, test, and debug code
- This fosters creativity
- Have many good minds work with innovative applications

Reasons for Participating in Open Source Communities

- Potential to reduce development costs
- Bolster their self-image
- Contribute something worthwhile to the software development community

Open Source Contribution and Differentiation

- Contributions to the open community and differentiation from the open community are for the following reasons:
  - Cost
  - Managing resources
  - Time it takes to bring a new product to the market

Reasons for Analyst Participation in the Open Source Community

- Curiosity about software benefits
- Achieve collective design
  - Incorporate open source software design into:
    - Proprietary products
    - Processes
    - Knowledge
    - IT artifacts

Collective Design

- Through a process of collective design the IT artifact is imbued with
  - Community and organizational structures
  - Knowledge
  - Practices

Summary

- Information is a key resource
- Integration of traditional systems with new technologies
- Roles and qualities of the systems analyst
- The systems development life cycle
- CASE tools
- Agile systems development
- Object-oriented systems development
- Open source systems


Any questions?
FUN GOPHER