6

Agile Modeling and Prototyping

Systems Analysis and Design, 8e

Kendall & Kendall

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Learning Objectives

- Understand the roots of agile modeling in prototyping and the four main types of prototyping.
- Be able to use prototyping for human information requirements gathering.
- Understand the concept of RAD for use in human information requirements gathering and interface design.
- Understand agile modeling and the core practices that differentiate it from other development methodologies.
- Learn the importance of values critical to agile modeling.
- Understand how to improve efficiency for users who are knowledge workers using either structured methods or agile modeling.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Agile Modeling, but First Prototyping

- Agile modeling is a collection of innovative, user-centered approaches to systems development.
- Prototyping is an information-gathering technique useful in seeking user reactions, suggestions, innovations, and revision plans.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Major Topics

- Prototyping
- Rapid application development (RAD)
- Agile modeling

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Prototyping

- Patched-up
- Nonoperational
- First-of-a-series
- Selected features

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Patched-Up Prototype

- A system that works but is patched up or patched together.
- A working model that has all the features but is inefficient.
- Users can interact with the system.
- Retrieval and storage of information may be inefficient.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Nonoperational Scale Models

- A nonworking scale mode that is set up to test certain aspects of the design.
- A nonworking scale model of an information system might be produced when the coding required by the application is too expensive to prototype but when a useful idea of the system can be gained through prototyping of the input and output only.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

First-of-a-Series Prototype

- Creating a pilot
- Prototype is completely operational
- Useful when many installations of the same information system are planned
- A full-scale prototype is installed in one or two locations first, and if successful, duplicates are installed at all locations based on customer usage patterns and other key factors.

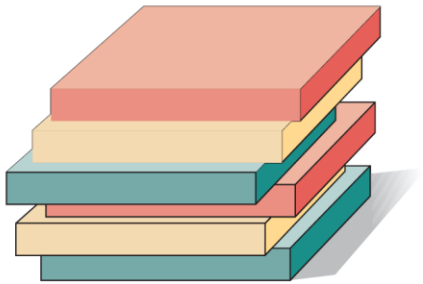
Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Selected Features Prototype

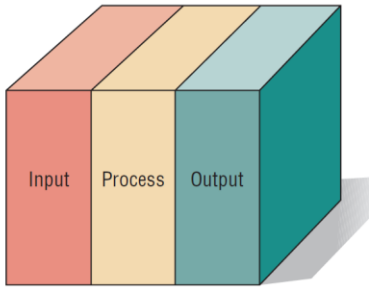
- Building an operational model that includes some, but not all, of the features that the final system will have
- Some, but not all, essential features are included
- Built in modules
- Part of the actual system

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

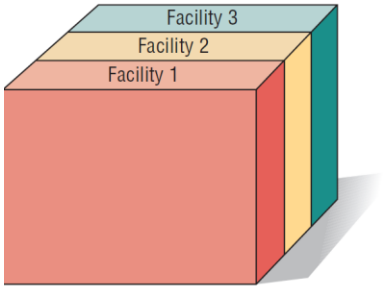
Four Kinds of Prototypes Clockwise, Starting from the Upper Left (Figure 6.1)



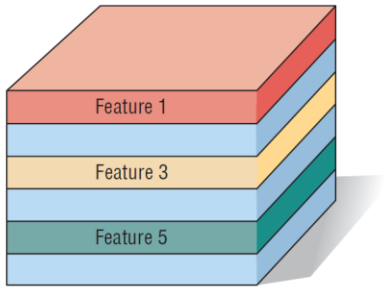
Patched-Up Prototype



Nonoperational Prototype



First-of-a-Series Prototype



Selected Features Prototype

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Prototyping as an Alternative to the Systems Life Cycle

- Two main problems with the SDLC
  - Extended time required to go through the development life cycle
  - User requirements change over time
- Rather than using prototyping to replace the SDLC use prototyping as a part of the SDLC

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Guidelines for Developing a Prototype

- Work in manageable modules.
- Build the prototype rapidly.
- Modify the prototype in successive iterations.
- Stress the user interface.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Disadvantages of Prototyping

- It can be difficult to manage prototyping as a project in the larger systems effort.
- Users and analysts may adopt a prototype as a completed system.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Advantages of Prototyping

- Potential for changing the system early in its development
- Opportunity to stop development on a system that is not working
- Possibility of developing a system that more closely addresses users' needs and expectations

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Prototyping Using COTS Software

- Sometimes the quickest way to prototype is through the modular installation of COTS software.
- Some COTS software is elaborate and expensive, but highly useful.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Users' Role in Prototyping

- Honest involvement
  - Experimenting with the prototype
  - Giving open reactions to the prototype
  - Suggesting additions to or deletions from the prototype

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Rapid Application Development (RAD)

- An object-oriented approach to systems development that includes a method of development as well as software tools

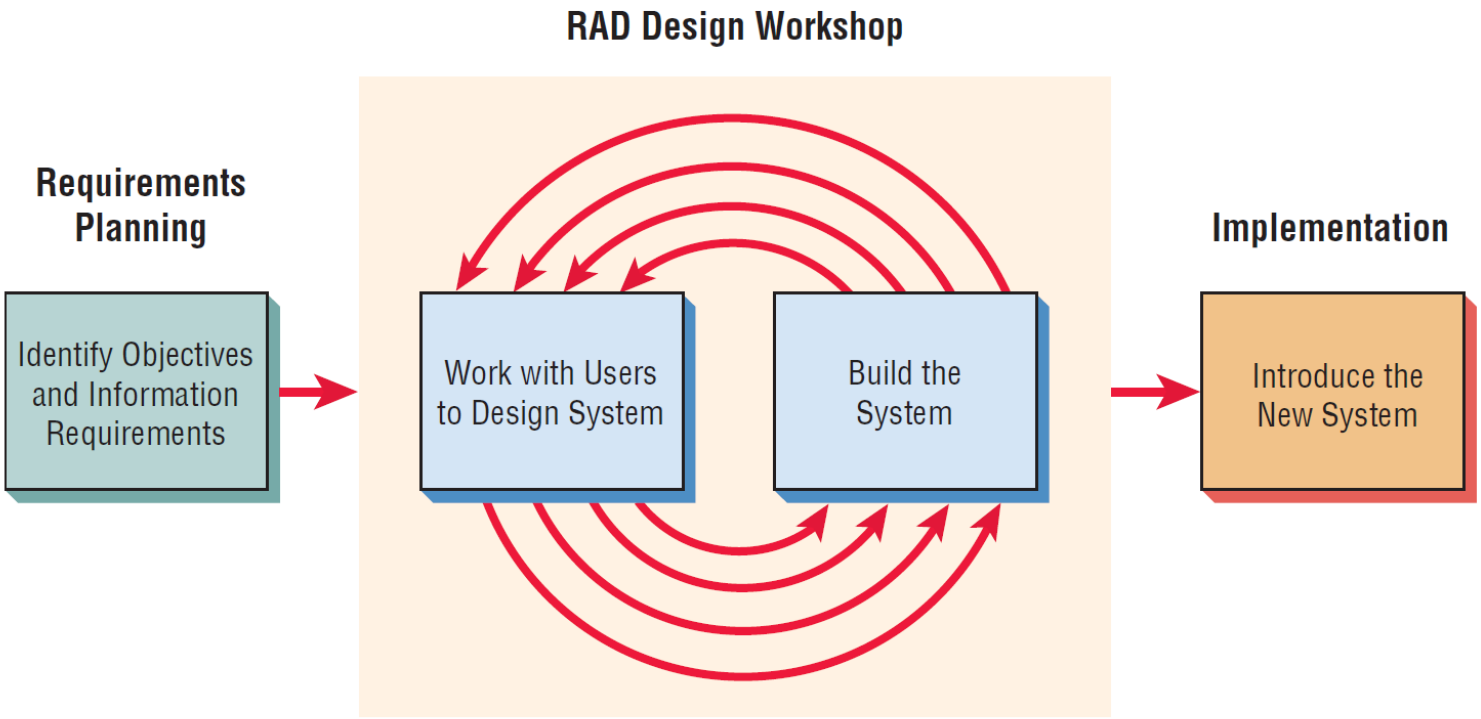
Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

RAD Phases

- Requirements planning
- RAD design workshop
- Implementation

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The RAD Design Workshop Is the Heart of the Interactive Development Process (Figure 6.4)



Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Requirements Planning Phase

- Users and analysts meet to identify objectives of the application or system.
- Orientation is toward solving business problems.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

RAD Design Workshop

- Design and refine phase.
- Use group decision support systems room if available.
- Users respond to actual working prototypes.
- Analysts refine designed modules based on user responses.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Implementation Phase

- As the systems are built and refined, the new systems or part of systems are tested and then introduced to the organization.
- When creating new systems, there is no need to run old systems in parallel.

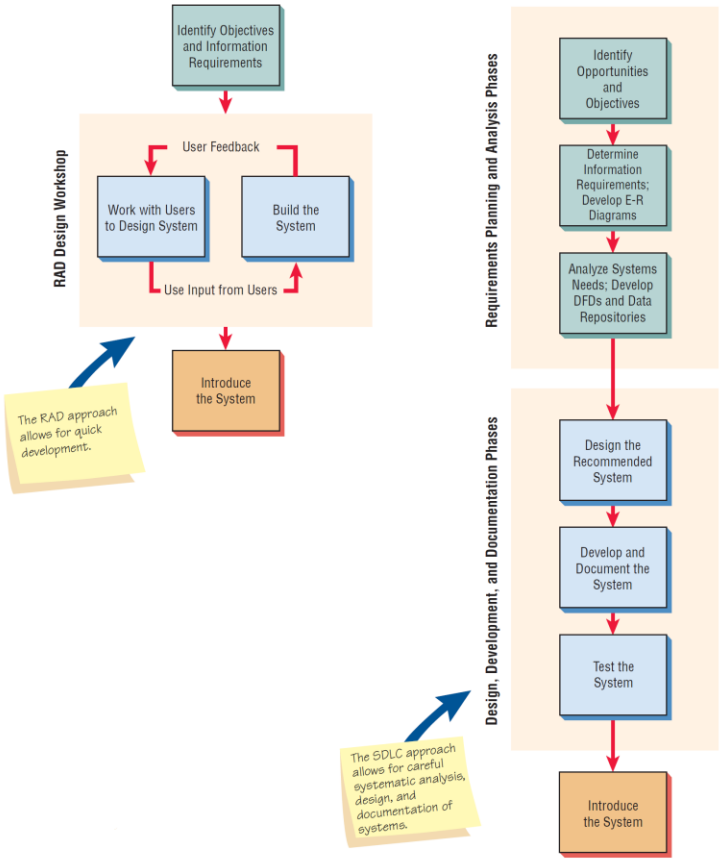
Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Comparing RAD to the SDLC

- RAD software tools are used to generate screens and exhibit the overall flow of the running of the application.
- RAD users are signing off on a visual model representation.
- RAD implementation is less stressful because users have helped to design the business aspects of the system.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The RAD Design Workshop and the SDLC Approach Compared (Figure 6.5)



Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

When to Use RAD

- The team includes programmers and analysts who are experienced with it.
- There are pressing reasons for speeding up application development.
- The project involves a novel ecommerce application and needs quick results.
- Users are sophisticated and highly engaged with the goals of the company.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Disadvantages of RAD

- Trying to hurry the project too much
- Lack of documentation

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Agile Modeling

- Agile methods are a collection of innovative, user-centered approaches to systems development.

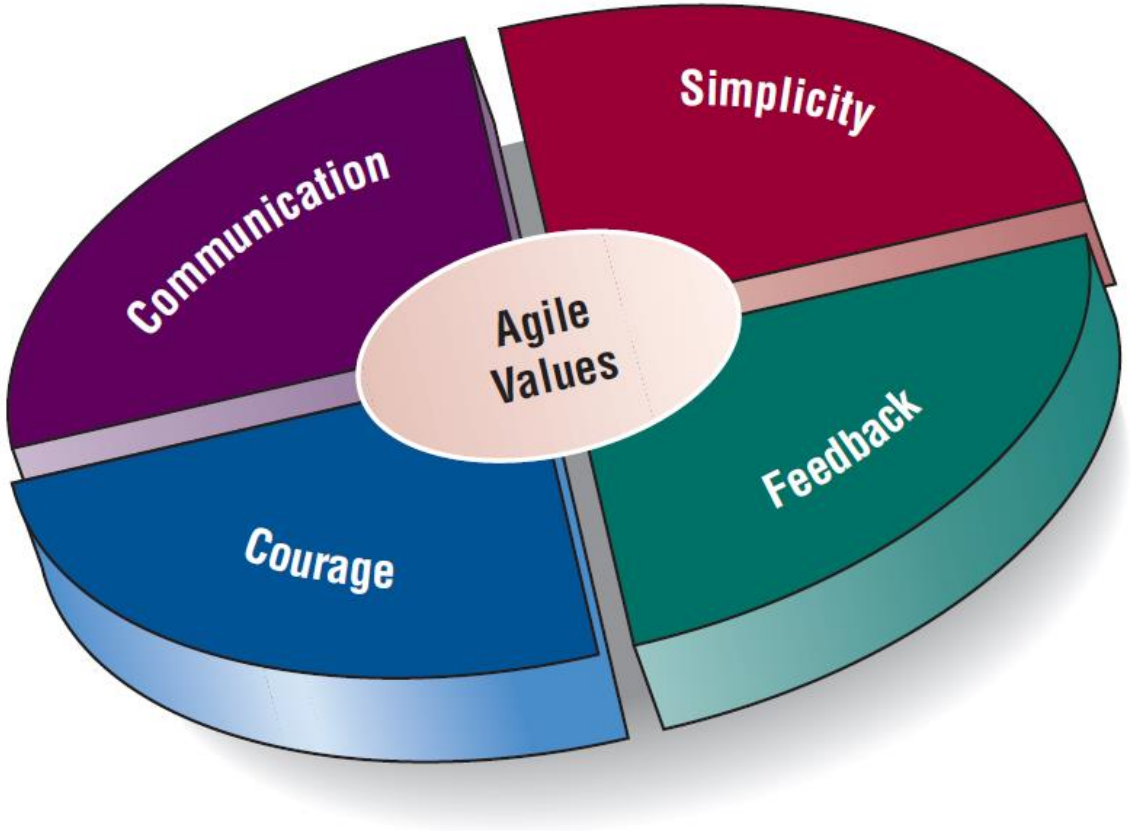
Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Values and Principles of Agile Modeling

- Communication
- Simplicity
- Feedback
- Courage

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Values Are Crucial to the Agile Approach (Figure 6.6)



Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The Basic Principles of Agile Modeling

- Satisfy the customer through delivery of working software.
- Embrace change, even if introduced late in development.
- Continue to deliver functioning software incrementally and frequently.
- Encourage customers and analysts to work together daily.
- Trust motivated individuals to get the job done.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The Basic Principles of Agile Modeling (continued)

- Promote face-to-face conversation.
- Concentrate on getting software to work.
- Encourage continuous, regular, and sustainable development.
- Adopt agility with attention to mindful design.
- Support self-organizing teams.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The Basic Principles of Agile Modeling (continued)

- Provide rapid feedback.
- Encourage quality.
- Review and adjust behavior occasionally.
- Adopt simplicity.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Activities, Resources, and Practices of Agile Modeling

- Coding
- Testing
- Listening
- Designing

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Four Resource Control Variables of Agile Modeling

- Time
- Cost
- Quality
- Scope

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Four Core Agile Practices

- Short releases
- 40-hour work week
- Onsite customer
- Pair programming

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

The Agile Development Process

- Listen for user stories.
- Draw a logical workflow model.
- Create new user stories based on the logical model.
- Develop some display prototypes.
- Create a physical data model using feedback from the prototypes and logical workflow diagrams.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Writing User Stories

- Spoken interaction between developers and users
- Seeking first and foremost to identify valuable business user requirements
- The goal is prevention of misunderstandings or misinterpretations of user requirements.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

User Stories Can Be Recorded on Cards: The User Story Should Be Brief Enough for an Analyst to Determine What Systems Features Are Needed (Figure 6.8)

Need or Opportunity:	Apply shortcut methods for faster checkout.					
Story:	If the identity of the customer is known and the delivery address matches, speed up the transaction by accepting the credit card on file and the rest of the customer's preferences such as shipping method.					
		Well Below	Below Average	Average	Above Average	Well Above
Activities:	Coding			✓		
	Testing			✓		
	Listening			✓		
	Designing					
Resources:	Time				✓	
	Cost		✓		✓	
	Quality					
	Scope			✓	✓	

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Scrum

- Begin the project with a high-level plan that can be changed on the fly.
- Success of the project is most important.
- Individual success is secondary.
- Project leader has some (not much) influence on the detail.
- Systems team works within a strict time frame.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Scrum

- Product backlog
- Sprint backlog
- Sprint
- Daily scrum
- Demo

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Lessons Learned from Agile Modeling

- Short releases allow the system to evolve.
- Pair programming enhances the overall quality.
- Onsite customers are mutually beneficial to the business and the agile development team.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Lessons Learned from Agile Modeling (Continued)

- The 40-hour work week improves worker effectiveness.
- Balanced resources and activities support project goals.
- Agile values are crucial to success.

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

There Are Six Vital Lessons that Can Be Drawn from the Agile Approach to Systems (Figure 6.9)

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Comparing Agile Modeling and Structured Methods

- Improving the efficiency of systems development
- Risks inherent in organizational innovation

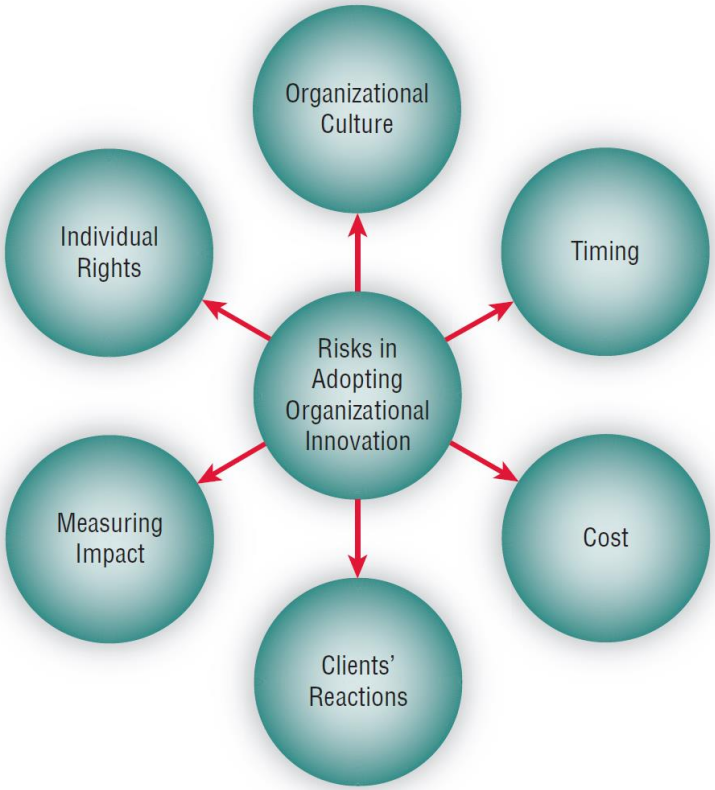
Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Strategies for Improving Efficiency Can Be Implemented Using Two Different Development Approaches (Figure 6.10)

Strategies for Improving Efficiency in Knowledge Work	Implementation Using Structured Methodologies	Implementation Using Agile Methodologies
Reduce interface time and errors	Adopting organizational standards for coding, naming, etc.; using forms	Adopting pair programming
Reduce process learning time and dual processing losses	Managing when updates are released so the user does not have to learn and use software at the same time	Ad hoc prototyping and rapid development
Reduce time and effort to structure tasks and format outputs	Using CASE tools and diagrams; using code written by other programmers	Encouraging short releases
Reduce nonproductive expansion of work	Project management; establishing deadlines	Limiting scope in each release
Reduce data and knowledge search and storage time and costs	Using structured data gathering techniques, such as interviews, observation, sampling	Allowing for an onsite customer
Reduce communication and coordination time and costs	Separating projects into smaller tasks; establishing barriers	Timeboxing
Reduce losses from human information overload	Applying filtering techniques to shield analysts and programmers	Sticking to a 40-hour workweek

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Adopting New Information Systems Involves Balancing Several Risks (Figure 6.11)



Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Risks When Adopting New Information Systems

- Fit of development team culture
- Best time to innovate
- Training cost for analysts and programmers
- Client's reaction to new methodology
- Impact of agile methodologies
- Programmers/Analysts individual rights

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Summary

- Prototyping
  - Patched-up system
  - Nonoperational
  - First-of-a-series
  - Selected-features
- Prototype development guidelines
- Prototype disadvantages

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Summary (Continued)

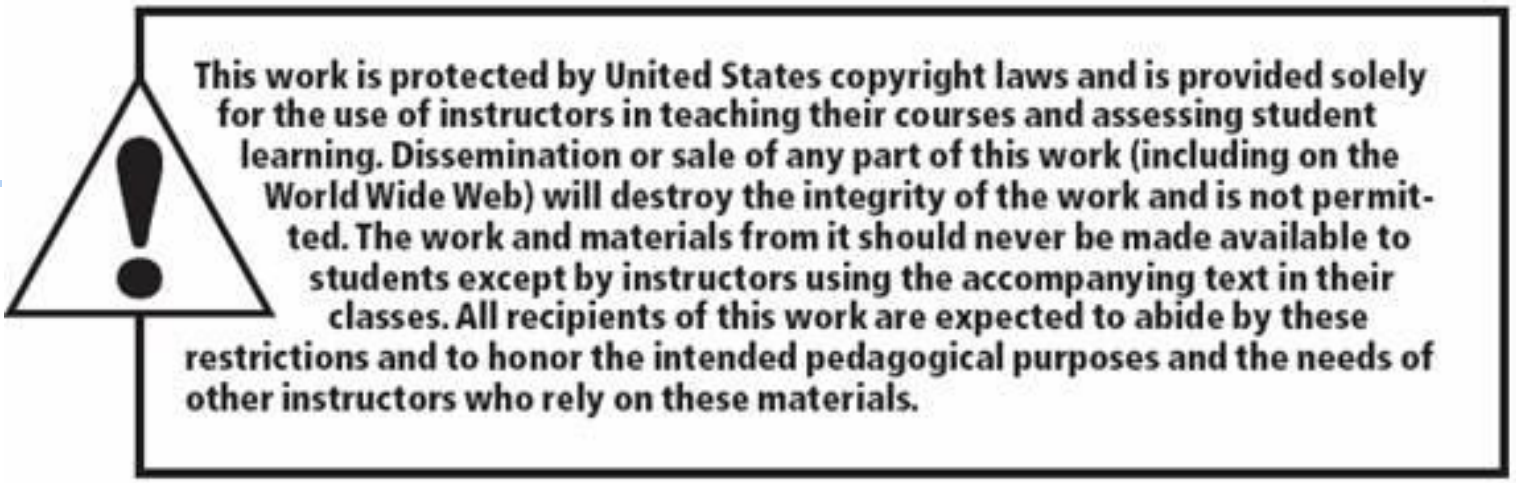
- Prototype advantages
- Users' role in prototyping
- Agile modeling
- Five values of the agile approach
- Principles of agile development
- Agile activities
- Agile resources

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall

Summary (Continued)

- Core practices of the agile approach
- Stages in the agile development process
- User stories
- Agile lessons
- Scrum methodology
- Dangers to adopting innovative approaches

Copyright © 2011 Pearson Education, Inc. Publishing as Prentice Hall



All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

Copyright © 2011 Pearson Education, Inc.
Publishing as Prentice Hall