

REVIEW LESSON

MTA Course: Software Development Fundamentals

Lesson name: Software Development Fundamentals 3.1

Topic: Understand application life cycle management (One 50-minute class period)

File name: SoftDevFund_RL_3.1

Lesson Objective:

3.1: Understand application life cycle management. *This objective may include but is not limited to:* phases of application life cycle management; software testing.

Preparation Details

Prerequisite student experiences and knowledge

Students should have had experience with multiple phases of the application life cycle and with testing software as units and as larger components. This MTA Certification Exam Review lesson is written for students who have learned about object-oriented programming. Students who do not have the prerequisite knowledge and experiences cited in the objective will find additional learning opportunities using resources such as those listed in the Microsoft® resources and Web links at the end of this review lesson.

Instructor preparation activities

None

Resources, software, and additional files needed for this lesson:

- SoftDevFund_PPT_3.1

Teaching Guide

Essential Vocabulary:

application lifecycle management (ALM)—a combination of business management practices and software engineering. ALM phases are Envisioning, Designing, Developing, Testing, and Maintenance.

test—to check program correctness by trying out various sequences and input values.

UML—acronym for Unified Modeling Language. A language developed by Grady Booch, Ivar Jacobson, and Jim Rumbaugh of Rational Software that can be used for specifying, building, and documenting software and non-software systems, such as business models. UML notation provides a common foundation for object-oriented design by providing descriptions of modeling concepts, including object class, associations, interface, and responsibility.

Lesson Sequence

Activating prior knowledge/lesson staging (5 minutes)

1. Show the Activator slide in the Microsoft PowerPoint® presentation for this lesson.
 - a. *Ask:* How are the software development lifecycle (SDLC) and application lifecycle management (ALM) related? How are they different?
Answer: ALM is more encompassing than SDLC. An application's lifecycle encompasses the whole time a company is spending money on a program, from the first idea to the end of the program's life. SDLC is actually just a component of ALM.

Lesson activity (25 minutes)

1. Show the PowerPoint presentation.
 - a. Review the definitions of ALM, test, and UML.
 - b. Discuss the benefits of an ALM model that promotes collaboration between the business of software and the engineering of software.
 - c. Discuss Envisioning as the first step in the ALM and how a business perspective can have benefits when it comes to developing the software.
 - d. Discuss design, and specifically, UML diagrams.
 - e. Discuss development.
 - f. Discuss testing—specifically, unit testing and regression testing.

- g. Explain the maintenance phase.
- h. Conclude by discussing how the ALM is not necessarily sequential or linear, but a general model for software production.

Assessment/lesson reflection (20 minutes)

1. Show the Lesson Review slide in the PowerPoint presentation.
2. Students should be given some time to write and share their answers with a peer. Call on students to share their answers with the class.
3. Students should illustrate the steps of the ALM, using an imaginary product as an example. They may draw a UML diagram for an imaginary project or for a recent class assignment.

Microsoft resources and Web links

Visual Studio Application Lifecycle Management (MSDN)

<http://msdn.microsoft.com/en-us/library/fda2bad5%28VS.100%29.aspx>

UML Component Diagrams (MSDN)

<http://msdn.microsoft.com/en-us/library/dd409393%28VS.100%29.aspx>

Suggested best practices:

- You might want to scaffold the lesson review by providing suggestions for the students' imaginary product.

Additional notes to the instructor:

- None