

UNIT 7 REVIEW

Key concepts in this unit

- Cost, schedule, quality tradeoff
- Activity graph, Gantt chart
 - Critical path
- Effort estimation: experts, records, models
 - Application points, productivity
- Risk management, impact, likelihood, exposure
- Test cases, test suites, whitebox vs blackbox, representative vs exhaustive
- Test-driven development
- Code review
- YAGNI, refactoring, bad smells

Check yourself

- Which of the following is *not* a step in using a COCOMO-like cost estimation model?
 - A. Rate the difficulty of each component
 - B. Rate how capable the programmers are
 - C. Rate the capabilities of the programmers' tools
 - D. Rate how annoying the customers are

Check yourself

- Which of the following is a good example of a CASE (computer-aided software engineering) tool?
 - A. Configuration management tools
 - B. Data modeling tools
 - C. Refactoring tools
 - ☒ D. All of the above

Check yourself

- The earliest that a certain activity can start is January 15. The latest that the activity can start (without delaying other activities) is March 15. What is the slack time?
 - A. 2 months
 - B. More than two months
 - C. Less than two months
 - D. Unknown – more information is required

Check yourself

- When do developers allocate time for testing?
 - A. Before writing any of the product's code at all
 - B. Mixed in while writing the product's code
 - C. After writing all of the product's code
 - ☒ D. It depends: Agile requires writing some code up front, and some code during the project, while waterfall mainly involves testing afterward

Check yourself

- What is the main difference between unit testing and integration testing?
 - A. Unit tests are automated; integration tests are not
 - B. Unit tests are expensive; integration tests are not
 - C. Unit tests cover single components; integration tests cover combinations of components
 - D. Unit tests are done by software engineers; integration tests are usually done by users

Check yourself

- Why perform representative testing?
 - A. Because we live in a representative democracy
 - ☒ B. Because testing every single possible input (exhaustively) is usually too expensive
 - C. Because representative inputs are the most difficult to imagine
 - D. Because representative testing is mandated by most customers

Check yourself

- What is the difference between whitebox and blackbox testing?
 - A. Whitebox testers can see the code they're testing; blackbox testers cannot
 - B. Whitebox testers are allowed to write code; blackbox testers are not
 - C. Whitebox testers use waterfall processes; blackbox testers use agile processes
 - D. None of the above

Check yourself

- Why perform regression testing?
 - A. Adding features can break old code
 - B. Eliminating requirements can add code
 - C. Firing programmers can cause them to steal code
 - D. None of the above

Check yourself

- Which method is most effective at finding bugs, in general?
 - A. Unit testing
 - B. Acceptance testing
 - ☒ C. Code inspection
 - D. Stakeholder review

Check yourself

- Succinctly, how does agile resolve the three-way tension among cost, time, and quality?
 - A. Plan, plan, plan. Always have a long-term plan.
 - ☒ B. Start simple and improve from there.
 - C. Invest, invest, invest. Cultivate relationships with venture capitalists.
 - D. None of the above

Check yourself

- Which agile rule has the highest precedence?
 - A. The system (code and tests together) must communicate everything you want to communicate.
 - B. The system must contain no duplicate code.
 - C. The system should have the fewest possible classes.
 - D. The system should have the fewest possible methods.

Check yourself

- Why refactor?
 - A. By manufacturing code a second time, refactoring helps to establish ownership.
 - B. By fixing parts of code that don't work right, refactoring reduces the number of bugs.
 - ☒ C. By reorganizing code without changing function, refactoring makes code more maintainable.
 - D. None of the above