Multiple-Choice Questions:

1. What often-false assumption does the waterfall model made about requirements specifications?

    a. Specifications are predictable

    b. Specifications are stable

    c. Specifications have low change rates

    d. All of the above

    e. None of the above

2. True or false? It is better to discover defects later in the development process. That way, you can have more of the system finished before you worry about fixing things.

    a. True

    b. False

3. An empirical process model iterates between…

    a. … feedback and adaptation

    b. … design and implementation

    c. … requirements gathering and design

    d. … user studies and testing

    e. None of the above

4. True or false? In iterative software development, it is recommended that iterations be 3 to 6 months in length.

    a. True

    b. False

5. If your project has unstable requirements (i.e., that are prone to change), you should use a waterfall process model.

    a. True

    b. False

6. In iterative development, how long should an iteration generally be?

    a. 1 week

    b. 2–6 weeks

    c. 2–4 months

    d. 6 months to a year

    e. None of the above

7. Which of the following is meant by a *software development process*?

    a. A running instance of a program; for example, a UNIX process is a softw. devel. process

    b. Something developers <u>do</u> to accomplish a goal during a project; for example, planning poker is a softw. devel. process for estimation

    c. Something developers <u>use</u> to accomplish a goal during a project; for example, Subversion is a softw. devel. process for configuration management

    d. A structure imposed on the development of a software product; for example, developing iteratively and incorporating best practices might be ingredients in a softw. devel. process

    e. None of the above

8. Which one of the these is a <u>bad</u> length for an iteration?

    a. 1 week
    b. 2 weeks
    c. 4 weeks
    d. 6 weeks
    e. All of the above

9. Which one of these <u>is</u> appropriate in an agile and iterative development process?

    a. Gather a complete set of requirements before designing/building anything.

    b. Implement the backend of the system first—that is, before implementing the frontend functionality with which users interact.

    c. Generate and maintain complete, detailed design documents, which comprehensively model all aspects of the design.

    d. Implement the system incrementally, building it up bit by bit.

    e. Test the code at the end, after the system has been completely implemented.

10. Which of the following is <u>not</u> an agile value?

    a. Individuals and interactions over processes and tools

    b. Working software over comprehensive documentation

    c. Customer collaboration over contract negotiation

    d. Responding to change over following a plan

    e. None of the above (i.e., all are agile values)

11. Which of the following problems does iterative development directly address?

    a. Design erosion

    b. Unstable requirements

    c. Program comprehension

    d. All of the above

    e. None of the above

12. Which type of process control model is appropriate for software development?

    a. A "defined" process control model

    b. An "empirical" process control model

    c. A "remote" process control model

    d. A "parallel" process control model

    e. None of the above

13. In the _____ development process, the various phases of development are completed sequentially, one after the other (e.g., gather all the requirements, then design the whole system, then implement the whole system, and so on).

    a. iterative

    b. waterfall

    c. agile

    d. spiral

    e. None of the above

14. True or false? In iterative software development, iterations should be 2 to 6 days in length.

    a. True

    b. False

15. In software engineering, defects that are discovered _____ are _____ to fix.

    a. earlier; more expensive

    b. later; more expensive

    c. by customers; more difficult

    d. by developers; more difficult

    e. None of the above

16. Which of the following is meant by a software engineering process?

    a. Tools, such as Git, that developers use to accomplish their software engineering goals

    b. An organization or structure imposed on the tasks and activities involved in developing a software product

    c. A thread of control in a multithreaded computing system, such as the Rails web server

    d. All of the above

    e. None of the above

17. Which term is best defined by the following text?

Development of a system through repeated cycles and in smaller portions at a time, allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system

    a. Configuration management

    b. Iterative development process

    c. Waterfall development process

    d. Verification and validation

    e. None of the above

18. Which term is best defined by the following text?

Development of a system whereby progress is seen as flowing steadily downwards through the phases of conception, analysis, design, construction, testing, production, and maintenance, and wherein one should move to a phase only when its preceding phase is reviewed and verified

    a. Configuration management

    b. Iterative development process

    c. Waterfall development process

    d. Verification and validation

    e. None of the above

19. Which of the following is an issue associated with the waterfall development process?

    a. Falsely assumes that requirements are stable and can be known from the start

    b. System defects are often discovered late in the development process

    c. Can lead to "analysis paralysis" wherein a considerable investment of time and effort is put into a project before any code is written

    d. All of the above

    e. None of the above

20. Following a _____ software engineering process tends to reveal defects early in development.

    a. Waterfall

    b. Iterative

    c. Sequential

    d. All of the above

    e. None of the above

Solutions:

1. d

2. b

3. a

4. b

5. b

6. b

7. d

8. a

9. d

10. e

11. b

12. b

13. b

14. b

15. b

16. b

17. b

18. c

19. d

20. b

Question:

Answer the following 3 related questions:

- What often-false assumption does the waterfall software engineering process make?
- Why does this false assumption cause considerable problems for waterfall?
- How does iterative development overcome these problems?

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Solution:

Waterfall software development makes the often-false assumption that requirements are mostly stable and can be known from the beginning.

This false assumption creates considerable problems for waterfall because the whole system may be developed before problems with the requirements are discovered. Furthermore, the later defects are discovered in a software product, the more expensive they are to fix (the Defect Cost Increase (DCI) Principle).

Iterative development overcomes these problems by maintaining a tight feedback loop. That is, feedback on the system is collected at regular intervals, revealing any problems early in the process when they are less expensive to correct.

Problem:

For each piece of text below, place an "I" and/or a "W" next to it if it corresponds to *Iterative* and/or *Waterfall* development process, respectively.

_____ A structure imposed on the development of a software product

_____ Development of a system through repeated cycles and in smaller portions at a time, allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system

_____ Development of a system whereby progress is seen as flowing steadily downwards through the phases of conception, analysis, design, construction, testing, production, and maintenance, and wherein one should move to a phase only when its preceding phase is reviewed and verified

_____ System defects are often discovered late in the development process

_____ Good if your project has unstable requirements (i.e., that are prone to change)

_____ Falsely assumes that requirements can be known from the start

_____ Can lead to "analysis paralysis" wherein a considerable investment of time and effort is put into a project before any code is written

_____ Has an "empirical" process control model

Solution:

I,W A structure imposed on the development of a software product

I Development of a system through repeated cycles and in smaller portions at a time, allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system

W Development of a system whereby progress is seen as flowing steadily downwards through the phases of conception, analysis, design, construction, testing, production, and maintenance, and wherein one should move to a phase only when its preceding phase is reviewed and verified

W System defects are often discovered late in the development process

I Good if your project has unstable requirements (i.e., that are prone to change)
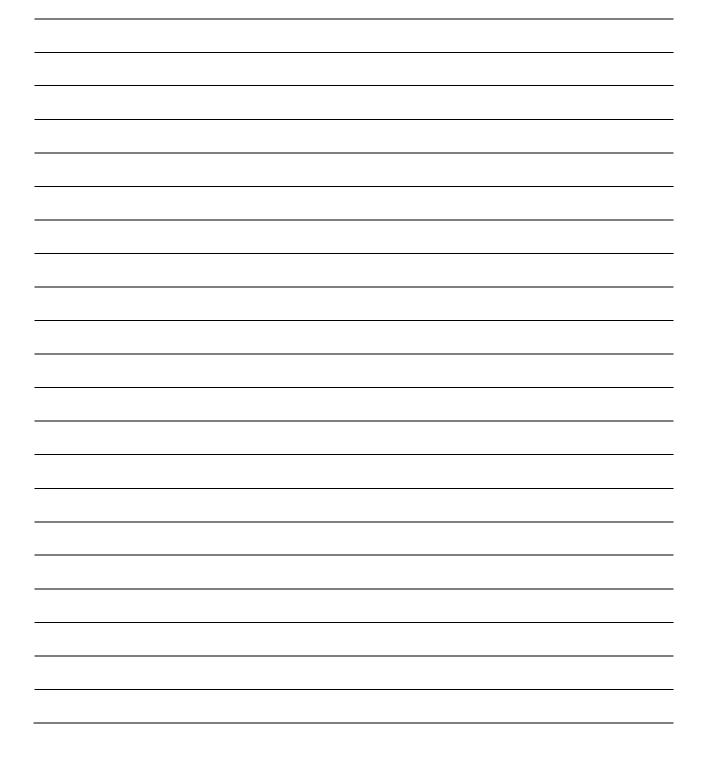
W Falsely assumes that requirements can be known from the start

W Can lead to "analysis paralysis" wherein a considerable investment of time and effort is put into a project before any code is written

I Has an "empirical" process control model

Question:

What are the key features of an empirical process model, does it effectively address unstable requirements (i.e., ones that are prone to change), and if so, how?

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Solution:

Key features of an empirical process model is that it iterates between feedback and adaption. Yes, it effectively addresses unstable requirements.

It does so via the feedback part of the process: as requirements change, the changes are revealed from the feedback. Because they are discovered quickly, they can be attended to early on when the cost of fixing them is lowest.

Feedback

Adaptation