Software Engineering

1. **Software is a product and can be manufactured using the same technologies used for other engineering** artifacts.
   a. True
   b. False

2. **Software deteriorates rather than wears out because**
   a. Software suffers from exposure to hostile environments
   b. Defects are more likely to arise after software has been used often
   c. Multiple change requests introduce errors in component interactions
   d. Software spare parts become harder to order

3. **Most software continues to be custom built because**
   a. Component reuse is common in the software world.
   b. Reusable components are too expensive to use.
   c. Software is easier to build without using someone else components.
   d. Off-the-shelf software components are unavailable in many application domains.

4. **Which of the following can be elements of software systems?**
   a. documentation
   b. b. software
   c. people
   d. hardware

5. **The functionality of most computer systems does not need to be enhanced in the lifetime of the system.**
   a. True
   b. False

6. **Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.**
   a. True
   b. False

7. **Most software development projects are initiated to try to meet some business need.**
   a. True
   b. False

8. **In general software only succeeds if its behaviour is consistent with the objectives of its designers.**
   a. True
   b. False

9. **Software engineers do not need to consider hardware when designing a computer-based system.**
   a. True
   b. False

10. **To construct a system model the engineer should consider which of the following restraining factors?**
    a. assumptions
    b. budget
    c. constraints
    d. schedule
11. **An example of the risk involved in software development is:**
    a. key personnel may resign before the product is complete.
    b. the manufacturer of critical components (e.g. the hardware associated with a real-time system) may go bankrupt.
    c. technology changes may render the product obsolete.
    d. competitors may market a fully functional lower-cost equivalent package.
    e. All of the above
12. **Software Engineering is best described as:**
    a. the practice of designing, building, and maintaining off-the-shelf software from prefabricated parts.
    b. the practice of designing, building and maintaining ad-hoc software without the use of formal methods.
    c. the practice of designing, building and maintaining reliable and cost-effective software using standard techniques.
    d. the practice of designing, building and maintaining fast and flexible software specifically for Engineering applications.
    e. the practice of designing, building and maintaining flashy, cheap and buggy software engineered to generate large initially sales and an on-going market for updates.
13. **What is the single largest computer-related cost for most organizations?**
    a. Software analysis and design.
    b. Software implementation.
    c. Software testing.
    d. Software maintenance.
    e. Coca Cola and pizza.
14. **A requirements specification is:**
    a. A rough list of things that the proposed software ought to do.
    b. A precise list of things that the proposed software ought to do.
    c. A formal list of things that the proposed software must do.
    d. A mathematical specification of the exact behaviour of the proposed software.
    e. An estimate of the resources (time, money, personnel, etc.) which will be required to construct the proposed software.
15. **The testing phase of software development doesn't require:**
    a. testing that the implementation compiles correctly.
    b. testing that the implementation matches the design.
    c. testing that the implementation matches the requirements.
    d. testing that the components of the implementation work separately and together.
    e. testing that the implementation interacts correctly with the environment.

16. **System maintenance is necessary because:**
    a. Humans never get it right the first time.
    b. The deployment platform may change over time.
    c. The user's needs may change over time.
    d. All of the above.
    e. None of the above.
17. **Maintenance may involve:**
    a. only additional coding and testing.
    b. only additional analysis and design.
    c. only additional design, coding and testing.
    d. any of the development phases, except analysis.
    e. any of the development phases.

Software Development Models

1. **Which of the following is not generally considered a player in the software process?**
    a. customers
    b. end-users
    c. project managers
    d. sales people
2. **The linear sequential model of software development is:**
    a. A reasonable approach when requirements are well defined.
    b. A good approach when a working program is required quickly.
    c. The best approach to use for projects with large development teams.
    d. An old-fashioned model that cannot be used in a modern context.
3. **The linear sequential model of software development is also known as the:**
    a. Fountain model
    b. Spiral model
    c. Waterfall model
    d. None of the above
4. **The incremental model of software development is:**
    a. A reasonable approach when requirements are well defined.
    b. A good approach when a working core product is required quickly.
    c. The best approach to use for projects with large development teams.
    d. A revolutionary model that is not used for commercial products.
5. **The spiral model of software development:**
    a. Ends with the delivery of the software product
    b. Is more chaotic than the incremental model
    c. Includes project risks evaluation during each iteration
    d. None of the above

6. **A simple way of looking at the spiral software life-cycle model is as a _________ model with each phase preceded by risk analysis.**
   a. Waterfall
   b. Incremental
   c. Synchronize and stabilize
   d. Formal methods
   e. Agile process
7. **In an analysis of some of the life cycle models, we can conclude that the __________ model is the best.**
   a. Waterfall
   b. Spiral
   c. XP
   d. Formal method
   e. Re-use
   f. None of the above.
8. **Process models are described as agile because they**
   a. eliminate the need for cumbersome documentation
   b. emphasize manoeuvrability and adaptability
   c. do not waste development time on planning activities
   d. make extensive use of prototype creation
9. **Software processes can be constructed out of pre-existing software process models to best meet the** needs of a software project.
   a. True
   b. False
10. **Which of the following is not one of the CBSE activities that take place for requirements that can be addressed with commercial off-the-shelf (COTS) components?**
    a. component adaptation
    b. component composition
    c. component design
    d. component qualification
11. **What are the two parallel engineering activities found the CBSE process model?**
    a. component-based development and library development
    b. domain engineering and component-based development
    c. domain engineering and process development
    d. none of the above
12. **In the most general sense a component is a modular building block for computer software.**
    a. True
    b. False
13. **Software engineers always need to create components from scratch in order to meet customer expectations fully.**
    a. True
    b. False

14. **Reusable software components must be**
    a. catalogued for easy reference.
    b. standardized for easy application.
    c. validated for easy integration.
    d. all of the above.
15. **A software process model is:**
    a. A representation of the way in which software is developed
    b. A representation of the way in which software processes data
    c. A representation of the way in which software is used
    d. A representation of the way in which software may fail
    e. An attractive young person used in the process of selling software
16. **The Waterfall Model is inadequate because:**
    a. Water is a continuous medium whereas code comes in discrete chunks (i.e. functions, objects, etc.), so all water-based analogies for software development are doomed to failure.
    b. it incorrectly suggests that the sequence of development is a progression from stage to stage, with no backwards steps.
    c. it incorrectly suggests that the sequence of development is a random process of rising and falling from stage to stage, with backwards progress just as likely as forwards.
    d. it incorrectly suggests that the sequence of development is a process unpredictable in the details, but predictable in its overall effect, like a waterfall.
    e. it incorrectly suggests that the sequence of software development is susceptible to uncontrollable external and internal forces (analogous to gravity and surface tension).
17. **The five general phases in the Spiral model are:**
    a. Analysis, Design, Implementation, Testing, and Review
    b. Review, Decision, Engineering, Acceptance, and Planning
    c. Analysis, Design, Engineering, Testing, and Payment
    d. Review, Risk-analysis, Prototyping, Engineering, and Planning
    e. Review, Risk-analysis, Design, Implementation, and Planning
18. **The Engineering phase of the Spiral model incorporates:**
    a. implementation only
    b. design and implementation
    c. analysis, design, and implementation
    d. analysis, design, implementation, and testing
19. **The Waterfall model of software development**
    a. Involves developing a series of prototypes
    b. Incorporates risk management
    c. Is considered the best way to develop software
    d. Suggests that one should perform the steps in a sequential manner without iterating.
    e. Does not allow one to correct any mistakes.

20. **Which of the following is not a disadvantage of the prototyping process model?**
    a. Not really a complete development methodology
    b. Product is not a complete system but may be treated as such by management
    c. Document is often sparse or completely absent
    d. Assumes a linear development approach

Software Requirements

1. **Which of the following is not captured by software requirements?**
   a. System services
   b. Interface with other systems
   c. System constraints
   d. Resource estimates
2. **During project inception the intent of the tasks are to determine**
   a. basic problem understanding
   b. nature of the solution needed
   c. people who want a solution
   d. none of the above

3. Three things that make requirements elicitation difficult are problems of
   a. budgeting
   b. scope c. understanding d. volatility

4. The work products produced during requirement elicitation will vary depending on the
   a. size of the budget
   b. size of the product being built
   c. software process being used
   d. stakeholder's needs

5. **It is relatively common for different customers to propose conflicting requirements, each arguing that his or her version is the right one.**
   a. True
   b. False
6. **Methods of requirements gathering include**
   a. interviews using structured techniques and close-ended questions
   b. interviews using structured techniques and open-ended questions
   c. questionnaires
   d. an analysis of forms used by clients
   e. All of the above.
7. **An advantage of scenarios is that they**
   a. demonstrate the behaviour of the product in a way that is comprehensible to the user.
   b. can be understood by the users and therefore the users play active roles throughout the requirement gathering process.
   c. play an important role in system analysis.
   d. may be depicted in a number of ways e.g. lists of actions, storyboards. e. All of the above.

8. **The system specification describes the**
   a. function, performance and constraints of a software system
   b. implementation of each allocated system
   c. element of software architecture
   d. time required for system simulation

9. **Requirements specifications that are written in a natural language such as English tend to have problems that include**
   a. Contradictions
   b. Ambiguities
   c. Omissions
   d. None of the above

10. **The best way to conduct a requirements validation review is to**
   a. examine the system model for errors
   b. have the customer look over the requirements
   c. ask the design team and see if they have any concerns
   d. use a checklist of questions to examine each requirement

11. **In requirements validation the requirements model is reviewed to ensure its technical feasibility.**
   a. True
   b. False

12. **The use of traceability tables in requirements engineering helps to**
   a. debug programs following the detection of run-time errors
   b. determine the performance of algorithm implementations
   c. identify, control, and track requirements changes
   d. none of the above

13. **A stakeholder is anyone who will purchase the completed software system under development.**
   a. True
   b. False

14. **The job of the requirements engineer is to categorize all stakeholder information in a way that allows decision makers to choose an internally consistent set of requirements.**
   a. True
   b. False

15. **An important aspect of prototyping is that**
   a. it is comprehensive
   b. it is built for change
   c. aspects such as error-checking capabilities, I/O and complex calculations are handled
   d. it evolves into the final product
   e. All of the above