**Subject Area Exam – Testing**

1. Why is regression testing an important part of any integration testing procedure?

Answer (Section 22.3.2):

The goal of integration testing is to make sure that independent modules that work correctly on their own do not interfere with one another when added to the same program (unforeseen side effects are always possible). Regression testing checks for defects propagated to other modules by changes made to an existing program.

1. Describe object-oriented unit testing.

Answer (Section 22.4.1):

Class testing for OO software is equivalent to unit testing for conventional software. The focus throughout is on designing and testing appropriate sequences of operations to exercise all class states. It makes little sense to test operations or algorithms individually for classes.

1. List four types of systems tests.

Answer (Section 22.8):

* Recovery testing
* Security testing
* Stress testing
* Performance testing

4. What are the attributes of a good software test?

Answer (Section 23.1):

* Has a high probability of finding an error
* Not redundant
* Should be capable of uncovering a whole class of errors
* Should not be too simple or too complex

5. Describe the differences between black-box testing and white-box testing.

Answer (Section 23.2):

Black-box testing involves testing the functionality of a software component without knowing the details of its internal logic. White-box testing involves testing the independent logic paths with full implementation knowledge.

6. What is equivalence partitioning as it applies to software testing?

Answer (Section 23.2):

A black-box testing technique in which the input domain is divided into classes of equivalent data items. Test cases are derived from combinations of elements from each equivalence class. Exhaustive testing of all input domain values is not necessary.

7. How does unit testing differ for object-oriented testing as compared to conventional software

unit testing?

Answer (Section 24.3.1):

You can no longer test a single operation in isolation (the conventional view of unit testing) but rather, as part of a class. Rather than testing an individual module, the smallest testable unit is the encapsulated class. Because a class can contain a number of different operations and a particular operation may exist as part of a number of different classes, the meaning of unit testing changes dramatically.

8. What is scenario-based testing?

Answer (Section 24.4.4):

The user tasks described in the use-cases are used to construct the test cases. It is used to uncover errors that occur when actors interact with the software (focus is on user behavior, not product behavior). Generally it is better to spend more time reviewing the use cases as they are created than spending more time on testing.

9. Describe the how test cases are derived from behavior models to facilitate interclass testing?

Answer (Section 24.6.2):

* Test cases must cover all states in the state transition diagram
* Breadth first traversal of the state model can be used (test one transition at a time and only make use of previously tested transitions when testing a new transition)
* Test cases can also be derived to ensure that all behaviors for the class have been adequately exercised

10. Summarize the steps used in a WebApp testing strategy.

Answer (Section 25.1.3):

* WebApp content model is reviewed to uncover errors.
* Interface model is reviewed to ensure all use-cases are accommodated.
* Design model for WebApp is reviewed to uncover navigation errors.
* User interface is tested to uncover presentation errors and/or navigation mechanics problems.
* Selected functional components are unit tested.
* Navigation throughout the architecture is tested.
* WebApp is implemented in a variety of different environmental configurations and the compatibility of WebApp with each is assessed.
* Security tests are conducted.
* Performance tests are conducted.
* WebApp is tested by a controlled and monitored group of end-users (looking for content errors, navigation errors, usability concerns, compatibility issues, reliability, and performance).

11. What are the objectives for content testing?

Answer (Section 25.3.1):

* Uncover syntactic errors in all media (e.g. typos)
* Uncover semantic errors (e.g. errors in completeness or accuracy)
* Find errors in organization or structure of content presented to end-user

12. What is the difference between load testing and stress testing?

Answer (Section 25.9):

Load testing attempts to determine how the WebApp and its server-side environments respond to various load conditions. Stress testing is a continuation of load testing that seeks to have the number of user, transactions, and data load meet and exceed operational limits.

13. List guidelines that should be followed in a MobileApp testing strategy.

Answer (Section 26.1):

* Understand the network landscape and device landscape.
* Conduct testing in uncontrolled real-world test conditions.
* Select the right automation test tool.
* Identify the most critical hardware/ platform combinations to test.
* Check the end-to-end functional flow in all possible platforms at least once.
* Conduct performance, GUI, and compatibility testing using actual devices.
* Measure MobileApp performance under realistic network load conditions.

14. List the quality attributes that should be tested for a MobileApp.

Answer (Section 26.2):

* Functionality
* Reliability
* Usability
* Efficiency
* Maintainability
* Portability

15. What is the testing in the wild for MobileApps?

Answer (Section 26.5):

Some of the characteristics of in-the-wild testing include: adverse and unpredictable environments, outdated browsers and plug-ins, unique hardware, and imperfect connectivity (both Wi-Fi and mobile carrier). In order to mirror real-world conditions, the demographic characteristics of your testers should match those of your targeted users, as well as matching their devices. In addition, you should include use-cases involving small numbers of users, less popular browsers, as well as a diverse set of mobile devices.

16. Which questions need to be answered in user-centered requirements elicitation?

Answer (Section 27.3.1):

* What are the users’ security software needs?
* How can a secure architecture be designed so it accommodates good user interface design?
* How can satisfying user experience be designed so that it is secure?

17. What are the elements of a security case?

Answer (Section 27.4.1):

* Security claims
* Arguments using evidence to link claims and assumptions
* Body of evidence supporting the arguments

18. What is an incidence response plan?

Answer (Section 27.6):

An incidence response plan spells out the actions to be carried out by each of the system’s stakeholders in response to specific security threats.