**Final Exam – Software Engineering Testing**

1. Explain what is wrong with the notion that computer software does not need to evolve over time.

Answer (Section1.1):

Computer software must be revised as errors are discovered and corrected. Software must be updated to accommodate changes in the computing environment. Many times a customer will request changes to add new functions to an existing product or to accommodate changes in the business environment. Sometimes an older system will need to be reengineered to provide benefits to the user in a modern context. The bottom line is that software that does not evolve will eventually become unusable.

1. Describe the relationships among software engineering process, methods, and tools.

Answer (Section2.1):

Software process defines the framework that must be established for effective delivery of software engineering technology, by providing a context by which the software engineering methods are applied. Software engineering methods provide the technical how-to’s for building software. Software engineering tools provide automated or semi-automated support for software engineering process and methods.

1. Why it important for software processes to be agile?

Answer (Chapter 3 Overview):

Software process provides the stability, control, and organization to an activity to prevent it from becoming chaotic. Yet, modern software processes must only demand the activities, controls, and work products that are appropriate for the team and product to be produced – to ensure that it can accommodate changes easily and deliver a high quality software product.

1. Why are evolutionary models considered by many to be the best approach to software development in a modern context?

Answer (Section 4.1.3):

Because time lines for the development of modern software are getting shorter and shorter, customers are becoming more diverse (making the understanding of requirements even harder), and changes to requirements are becoming even more common (before delivery), we need a way to provide incremental or evolutionary delivery. The evolutionary process accommodates uncertainty better than most process

1. Describe the three key assumptions regarding software projects that every agile software process must address.

Answer (Section 5.3):

It is difficult to predict in advance which software requirements and customer priorities will change and which will not.

For many types of software design and construction must be interleaved, it is difficult to predict how much design is needed before construction can be used to prove the design.

Analysis, design, construction, and testing are not always predictable processes and this makes planning difficult.

1. List environment characteristics that can be considered toxic to software teams.

Answer (Section 6.3):

Frenzied work atmosphere

High frustration that causes friction among team members

Fragmented or poorly coordinated software process

Unclear definition of roles on the software team

Continuous and repeated exposure to failure

Differing and incompatible team member human traits

1. What work products result from the requirements engineering process?.

Answer (Section 8.1):

The intent of requirements engineering is to provide stakeholders with a written understanding of the problem, the work products produced include usage scenarios, function and feature lists, and requirements models

1. What is a formal technical review and why is one conducted? Outline the steps required to conduct a successful FTR?

Answer (Section 20.6):

The purpose of an FTR is to have a group of software engineers examine a discrete work product and determine whether on not the product is free of defects using the software specifications and standards as the review criteria.

To perform a successful FTR, the steps described in Section 15.6.3 are conducted.

1. What are the key differences between validation testing goals and acceptance testing goals?

Answer (Section 22.7):

In validation testing, the test team seeks to ensure that each software function or performance characteristic conforms to its specification. In acceptance testing, the test team needs to ensure that the software works correctly for the intended user in his or her normal work environment.

1. Describe three control structure testing strategies.

Answer (Section 23.5):

Condition or branch testing -uses test cases that exercise every decision statement in the program.

Data flow testing - selects test paths (definition use chains) according to the locations of variable definitions and uses in the program

Loop testing -tests focus on the validity the repetition constructs (making sure that loops start and stop when they are supposed to)

1. Describe three partitioning strategies that can be used when performing class level testing for OO systems.

Answer (Section 24.5.2):

State-based partitioning - tests designed so that operations that cause state changes are tested separately from those that do not

Attribute-based partitioning - for each class attribute, operations are classified according to those that use the attribute, those that modify it, and those that do not use or modify the attribute

Category-based partitioning - operations are categorized according to the function performed: initialization, computation, query, or termination

1. Describe the WebApp interface testing strategy.

Answer (Section 25.4.1):

* Interface features are tested to ensure that design rules, aesthetics, and related visual content is available for user without error.
* Individual interface mechanisms are tested using unit testing strategies.
* Each interface mechanism is tested in the context of a use-case of navigation semantic unit (e.g. thread) for a specific user category
* Complete interface is tested against selected use-cases and navigation semantic unit to uncover interface semantic errors
* Interface is tested in a variety of environments to ensure compatibility

1. What types of testing should be followed in a comprehensive MobileApp testing strategy.

Answer (Section 26.2):

* Conceptual Testing
* Unit and System Testing
* User Experience Testing
* Stability Testing
* Connectivity Testing
* Performance Testing
* Compatibility Testing
* Security Testing
* Certification Testing

1. Describe steps needs to create a threat model.

Answer (Section 27.5):

* Identify assets
* Create architectural overview
* Decompose application
* Identify threats
* Document threats
* Rate threats

15. Describe the process of writing a formal specification for some system function.

Answer (Section 28.6):

* Define a system state in terms of the objects manipulated by the function
* Using a simple mathematical function as an example, define the data invariant by writing data relations that will not change during the execution of the function
* Write the precondition and postcondition for the function using mathematical notation to show the system state before and after the

1. Describe the change control process for a modern software development project.

Answer (Section 29.3.3):

A change request is submitted for evaluation for a change report is submitted to the change control authority (CCA). The CCA makes the final determination as to the status and priority of the change. An engineering change order (ECO) is generated for each approved change. Items to be changes are checked out of the project database subject to its access control parameters. The modified object is subjected to SQA procedures and returned to the project database. Version control procedures are followed to produce the next version of the software. Synchronization control is used to make sure that parallel changes made by different people do not overwrite one another.

1. How does software configuration management differ for Web and Mobile Apps?

Answer (Section 29.4)

The “code and go” philosophy dominates App development. So SCM for Apps must be an agile process. Documentation and review of changes is done on an as needed basis depending on the risk associated with the work products being changed.

1. Describe the five activities associated with the software measurement process.

Answer (Section 30.1.3):

Formulation - derivation of metrics and measures that are appropriate for the software representation being considered

Collection - mechanism used to gather the data used to derive the metrics

Analysis - metric computation and use of mathematical tools

Interpretation - evaluation of metrics results to gain insight into the quality of the software representation

Feedback - recommendations derived from the interpretation are transmitted to the software team

1. Describe the role of class-oriented metrics in assessing the quality of an OO system.

Answer (Section 30.3.3):

The class is the fundamental unit of an OO system. The number of methods and their complexity are directly related to the effort required to test a class. The depth of the inheritance tree can be used to estimate the complexity of the class hierarchy. It is important to strive to have low coupling between classes and high cohesion within each class. If large class sizes are detected during OOA review, the classes should partitioned to improve the modularity of the system and make it easier to maintain.

1. Describe the change control process for a modern software development project.

Answer (Section 29.3.3):

A change request is submitted for evaluation for a change report is submitted to the change control authority (CCA). The CCA makes the final determination as to the status and priority of the change. An engineering change order (ECO) is generated for each approved change. Items to be changes are checked out of the project database subject to its access control parameters. The modified object is subjected to SQA procedures and returned to the project database. Version control procedures are followed to produce the next version of the software. Synchronization control is used to make sure that parallel changes made by different people do not overwrite one another.