**COSC 4317 Software Engineering Test 3 Spring 2017 Burris**

***Answer two of the following including question number 1****!*   *A professional should be able to answer every question*. Number your questions on the answer sheets in ascending numeric order from one through four inclusive. Write delete on the *answer sheets* by the two questions you do not wish graded. You may delete or answer more than one question on a page. Leave at least a one-inch margin at the top of every page. Do not write on the back of pages. Staple your answer sheets (in ascending numeric order) on top of the test in the upper left hand corner. Write your name in the upper right hand corner of the answer sheets (first page). Turn the stapled bundle over and write your name in the upper right hand corner on the back of the test. A five point Road Map Fee (RMF) will be deducted for each instance of failure to follow instructions. You will not receive credit for material I cannot read or that is obstructed from my view (especially if under the staple).

Warning: There are no “short” answers on this test. Tell me everything germane to the topic. You should be able to write 10 or more pages of pertinent information.

1. Discuss the 7 methods covered in class to optimize packaging for either a paged or overlay virtual memory system but not both. You must convenience me you know how to physically accomplish the optimizations in the selected virtual memory implementation. **Do not cover** motivations, when the packaging should occur, special considerations for virtual memory systems, optimization by language translators, language specific optimizations or other related topics.
2. State the most important topic covered by DeMarco and Lister in “Peopleware” and discuss it in detail. You must support your assertion it was the most important topic! Your question must not relate to another question on the test.
3. The most important technique covered in class to date with respect to producing high quality software this semester is called a “Structured Walk Through” or “Technical Review.” First explain why this is true. Second discuss how the walk through is accomplished and when in detail.
4. Discuss “white box” testing in detail including McCabe’s Measure.

Section 1: Answer one question.

1. Discuss optimization for time in detail emphasizing the 5 step method developed in class. You must clearly state when optimization should cease as you do not anticipate meeting optimization goals and how you made the decision.
2. Discuss the relationship between schedule, functionality and quality using Fredrick Brooks and Larry Putman’s material as a guide. You must evaluate each component in detail to receive credit. Do not select this option unless you can provide more than a page of pertinent information.
3. Discuss the results of 1985 Survey by Lawrence and Jeffrey and the Metric Premise by Weinberg-Schulmam (“Human Factors,” 1974, Vol. 16, pp 70-77). You must convenience me you read the material and mastered the conclusions. Do not select this option unless you can provide more than a page of pertinent information.
4. Discuss “Flow” in detail.

Section 2: Answer one question.

1. The most important technique covered in class to date with respect to producing quality software this semester is called a “Structured Walk Through” or “Technical Review.” First explain why this is true. Second discuss how the walk through is accomplished and when.
2. Discuss “white box” testing in detail including McCabe’s Measure.
3. Discuss statistical Quality Assurance (SQA) in detail including the Pareto principle and work of Robert Grady and Deborah Caswell.
4. Discuss developing a test plan in detail. What should be used to develop a plan. What and how should testing be accomplished. Include the specification, design, code and other pertinent information in your response.
5. First explain how a Structured Walk Through is conducted. Second explain why it may be the most important tool available to improve software quality.
6. Choice (do only one of the following):
7. Discuss statistical Quality Assurance (SQA) in detail including the Pareto principle and work of Robert Grady and Deborah Caswell.
8. State and explain the four methods discussed in class to produce software capable of detecting a run time error has occurred.
9. Discuss in detail “Black Team” methodology, management support, utilization, and why they tend to be very successful. A very detailed response with project specific information is expected. Include all areas from which team should be expected to extract test cases and how many test cases they should develop based on program logic.
10. Discuss either decision tables or finite state automata (not both) in detail stating their advantages and disadvantages. Be sure to include how they can be used to minimize the cost of making software changes due to a changing user environment.
11. Discuss optimization issues specific to overlay and virtual memory systems in detail with respect to space. Relate cohesion, coupling and black boxes to minimizing page/overlay faults. Emphasize optimization for space, not the differences between virtual paged memory and overlay schemes. *Hints: program structure, looping, code duplication, code factoring/module, binary search, fan-in, programming language structures that support minimizing page/overlay faults, etcetera.*
12. As you know the push to colonize the new territories has placed a tremendous strain on our resources. It is imperative that you create software to allow us to track all shuttles, X-wing fighters and space stations from a central control facility to be created in each sector of colonized space. Each shuttle, X-wing and space station will be required to identify itself on demand, give its location/destination and present course to the central control facility. Shuttles and space stations must accept course adjustments on demand from the central control facility. Each space station, shuttle, and X-wing have a unique serial number and alphanumeric name to a maximum of 20 characters in length. X-wings are launched and retrieved from space stations. Shuttles are used to transfer personnel and equipment between the central control facility and space stations. Space stations have photon torpedoes and proton guns. X-wings are limited to proton guns. X-wing fighters must protect all other entities from alien vessels which also have proton guns. Shuttles have cargo bays and no armaments. X-wing fighters and shuttles use ion propulsion systems using the same methods to start and stop their engines. All entities utilize the same life support and navigation systems. Space stations use warp engines and require a different procedure to start their engines. **Implement the Object Model in OOD using OMT including an inheritance hierarchy to reduce implementation cost while increasing reliability. You may but are not required to submit timing diagrams or state model.**
13. DeMarco and Lister are well known for “Coding War Games.” Do not discuss the war games. Rather discuss the results emphasizing what their studies showed does improve productivity and quality versus what does not improve productivity and quality. What are the lessons to be learned by industry with respect to staffing (hiring), attracting the best talent, and range of anticipated performance?
14. Discuss optimization issues specific to overlay and virtual memory systems in detail with respect to space. Relate cohesion, coupling and black boxes to minimizing page/overlay faults. Emphasize optimization for space, not the differences between virtual paged memory and overlay schemes. *Hints: program structure, looping, code duplication, code factoring/module, binary search, fan-in, programming language structures that support minimizing page/overlay faults, etcetera.*
15. Discuss the 5 step organized process we developed in class to optimize a system for time in detail. You discussion should include appropriate examples, motivations, and required information for each step of the process.
16. First explain the difference between multiple-inheritance in languages like C++ and Small Talk versus multiple-inheritance in Java. Next explain how you would solve the following problem using languages like Java (there is more than one possible solution technique) which only support single inheritance. You must point out and explain any advantages and disadvantages of your solution relative to an implementation using languages supporting multiple inheritance like C++ and Small Talk. Clearly identify any meta-classes in your solution. The emphasis here is on design and implementation technique not actual code. You need not write any actual code.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Carnivore |  |  |  |  | Mammal |  |  |  |
|  | Data Structures |  |  |  |  | Data Structures |  |  |  |
|  | KillPrey( )  EatPrey( ) |  |  |  |  | LoveMate( )  ProtectHome( ) |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  | Cat |  |  |  |  |  |  |
|  |  |  | Data Structures |  |  |  |  |  |  |
|  |  |  | Speak( ) |  |  |  |  |  |  |

1. The customer has provided us with the following design. **They require us to create a heterogeneous array holding 20 items which may be of type Jet, Propeller, Car or Truck**.

**Choice (do only one of the following):**

1. Explain in detail at least one option (preferably two options) to solve the problem (you may utilize the classes in the following design without coding them). Hint: Object
2. Write a minimum amount of Java code to create the heterogeneous array. Place at least one Jet and one Truck in the array. You may assume the classes have been implemented including constructors (you may utilize the classes in the following design without coding them).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Class Plane |  |  |  | Class Vehicle |  |
|  | Data |  |  |  | Data |  |
|  | Methods |  |  |  | Methods |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Class Jet |  | Class Propeller |  | Class Car |  | Class Truck |
| Data |  | Data |  | Data |  | Data |
| Methods |  | Methods |  | Methods |  | Methods |

1. Discuss prototyping in great detail. Your discussion should include the circumstances under which prototyping is most likely to be the best software development paradigm. Include the circumstances under which prototyping is not particularly useful and what should be used.
2. First discuss the purpose for coding war games from the standpoint of the authors of “Peopleware.” What did they hope to accomplish? Next state the traditional items people felt affected the productivity and quality of software projects. For each item explain the conclusions reached by De Marco and Lister.
3. Circle the letter preceding all correct statements pertaining to “major goals of OOD and OOP.”
4. Minimize the expense in time, cost, and other resource requirements during original implementation.
5. Maximize re-use of the software components.
6. Minimize the cost maintenance, fixing bugs.
7. Minimize the cost of modification, responding to changes in the environment.
8. Maximize the efficiency of resource utilization.
9. Maximize the ability to optimize software with respect to the hardware/software environment while minimizing the cost of the change.
10. Allow for extension of the language by a user adding abstract data type (ADT’s).
11. Circle the letter preceding all correct statements pertaining to OOD and OOP.”
12. Modify the Body Only: If you modify only the body then in principle it should be possible to recompile the body and link it to the specification. This would automatically update all users without further effort.
13. Modify the Specification: The specification, body, and all user code must be recompiled and linked when data structures or functionality is added to the specification. It will not be necessary to modify any user code however unless the user code wishes to take advantage of new functionality defined in the interface. If functionality is removed from the specification, or the interface for invoking a method is modified, then the user code that exercises these features will also have to be modified.
14. Modification of User Code: Modification of user code should not affect the way objects are utilized by the rest of the system.
15. OOP Modification: Any time the specification or body of a class is modified, the specification, body, and all code utilizing the class must be recompiled and linked to prevent errors.

Briefly explain each of the following with respect to OO specification, design, and programming.

***ABSTRACTION:***

**The essence of abstraction is to extract essential properties while omitting inessential details.**

***MODULARITY:***

**Modularity applies to the physical architecture of the system. It is purposeful structuring to help manage complexity.**

***LOCALIZATION:***

**Localization is primarily concerned with physical proximity, a collection of related data structures and operations on these data structures are collected into a single highly cohesive unit with minimal coupling to the rest of the system.**

***UNIFORMITY:***

**Uniformity simply means that the modules are free from any necessary differences, it usually results from a consistent notation, consistent control structures, and calling sequences.**

***COMPLETENESS:***

**Completeness ensures that all important elements (data and functions) are present.**

***CONFIRMABILITY:***

**”Confirmability implies that we must decompose our system so that it can be readily tested thus helping to make systems more easily modified and maintained.**

* 1. **Select any 4 of the above and explain in detail how they are related to metrics we have developed previously including Miller’s Law, Graicunas Law, coupling, cohesion, scope-of-effect/scope-of-control, highly factored, etcetera.**

1. Discuss prototyping in great detail. Your discussion should include the circumstances under which prototyping is most likely to be the best software development paradigm. Include the circumstances under which prototyping is not particularly useful and what should be used.
2. Choice (do only one of the following):

A) Discuss “Formal Technical Review” sometimes referred to “Structured Walk Through” in detail. Explain how they should be done, when, by whom, and how effective they are based on empirical evidence.

1. Glen Meyers established a set of five goals in "The Art of Software Testing," (Wiley 1979) to be achieved when developing software test cases. Please state and explain the goals. Your discussion should include McCabe’s “Cyclomatic Complexity Measure” and “Basis Path Testing” in detail. How would McCabe’s techniques be employed in industry.
2. Compare and contrast “White Box” and “Black Box” testing in detail. State their goals explaining how these goals are achieved.
3. Choice (do only one of the following):
4. DeMarco and Lister are well known for “Coding War Games.” Do not discuss the war games. Rather discuss the results emphasizing what their studies showed does improve quality and performance. Include what does not or only minimally affects quality and performance and why. What are the lessons to be learned by industry with respect to staffing (hiring), attracting the best talent, and range of anticipated performance?
5. Critique the following statements by DeMarco and Lister. You must support your position utilizing material from their book. You need not agree with them but you must convenience me you have a detailed knowledge of their point of view.

**Maximize the amount of delivered function (weighted by years of useful system life) per dollar of total system lifetime cost.**

**The default goal for projects that are not monitored for all development and quality aspects is “Completion in the shortest possible time.” Time is the only variable directly observable by management that does not require special effort!**

**Control of a software project requires measurement of project activities and deliverables!**

***The major goal for most software managers should not be goal attainment, but rather “Goal Alignment.”***

**The proper function of management is not to make people work, but to make it possible for them to work!**

4) Discuss all facets of McCabe’s “Cyclomatic Complexity Measure” and “Basis Path Testing” in detail. How would McCabe’s techniques be employed in industry.

1. Choice (do only one of the following):
2. Discuss Parkinson’s Law, the survey by Lawrence and Jeffery, and Metric premise by Weinberg-Schulman in detail.
3. Fredrick Brooks and Larry Putman state the tradeoffs between schedule, functionality, and quality is not linear. Assume we cannot make the required deadline. The following have been suggested as possible solutions to meet schedule requirements. Discuss the tradeoffs in detail. What will be the most likely outcome based on these suggestions and explain why in detail. A) Add personnel to the project. B) Reduce functionality. C) Reduce product quality.
4. Choice (do only one of the following):
5. Discuss “Formal Technical Review” sometimes referred to “Structured Walk Through” in detail. Explain how they should be done, when, by whom, and how effective they are based on empirical evidence.
6. Glen Meyers established a set of five goals in "The Art of Software Testing," (Wiley 1979) to be achieved when developing software test cases. Please state and explain the goals.
7. Choice (answer only one of the following):
8. Critique the following statements by DeMarco and Lister. You must support your position utilizing material from their book. You need not agree with them but you must convenience me you have a detailed knowledge of their point of view.

**Maximize the amount of delivered function (weighted by years of useful system life) per dollar of total system lifetime cost.**

**The default goal for projects that are not monitored for all development and quality aspects is “Completion in the shortest possible time.” Time is the only variable directly observable by management that does not require special effort!**

**Control of a software project requires measurement of project activities and deliverables!**

***The major goal for most software managers should not be goal attainment, but rather “Goal Alignment.”***

**The proper function of management is not to make people work, but to make it possible for them to work!**

1. Frederick Brooks states there is an important relationship between schedule (delivery date), functionality (feature richness), and quality (absence of defects) in software projects. Discuss his hypothesis in detail (at least 2 pages). How does it pertain to the material discussed by DeMarco and Lister.

C) Discuss the material covered by DeMarco and Lister in detail. Pay particular attention to improving quality and quantity while minimizing development time. Be very specific as to details! I am looking for facts as opposed to generalizations.

1. Choice (Answer only one of the following in great detail):
2. Compare and contrast prototyping, the water fall model, and the spiral model in detail.
3. Explain in detail the circumstances under which prototyping is most likely to be the best software development paradigm. Conversely, when is it not needed and what should be used?
4. Choice (answer only one of the following in great detail):
5. Compare and contrast two of the following team organizations: traditional, Chief Programmer, Weinberg, and Agile.
6. Discuss “Formal Technical Review” sometimes referred to “Structured Walk Through” in detail. Explain how they should be done, when, by whom, and how effective they are based on empirical evidence.
7. Choice (answer only one of the following in great detail):
8. Compare and contrast “White Box” and “Black Box” testing in detail. State their goals explaining how these goals are achieved.
9. Discuss all facets of McCabe’s “Cyclomatic Complexity Measure” and “Basis Path Testing” in detail. How would McCabe’s techniques be employed in industry.
10. Choice (do only one of the following):
11. Discuss Parkinson’s Law, the survey by Lawrence and Jeffery, and Metric premise by Weinberg-Schulman in detail.
12. Fredrick Brooks and Larry Putman state the tradeoffs between schedule, functionality, and quality is not linear. Assume we cannot make the required deadline. The following have been suggested as possible solutions to meet schedule requirements. Discuss the tradeoffs in detail. What will be the most likely outcome based on these suggestions and explain why in detail. A) Add personnel to the project. B) Reduce functionality. C) Reduce product quality.
13. Discuss “Decision Tables” in detail including advantages and implementation techniques.

Discuss flow, subliminal suggestion, and measures of software quality in detail.

1. The definition of what constitutes acceptable quality may vary with circumstances, e.g., sales volume, profit margin per unit, etcetera. Discuss the concepts of “quality beyond what the user requires,” “acceptable quality,” “builder/manager/customer” views of quality, and the “Flight from Excellence” in detail based on Demarco and Lister.
2. Four methods were discussed in class for developing software that can detect and report errors it commits. List the four methods and explain them in detail.
3. Discuss “optimization for time” in detail in a paged memory environment. You must explicitly explain how to determine where optimization should occur and a method to prioritize the optimization process.
4. Discuss optimization for space in great detail with respect to overlay schemes. You must be very specific with respect to physically accomplishing the optimization process.
5. Discuss Object Oriented Design (OOD) in great detail. Include the 3 models most design will require and what is included in great detail. You must clearly explain how the models are related.
6. Explain in great detail the meaning of public, protected, and private modifiers with respect to inheritance with examples. You may use our Object Oriented Design (OOD) notation (OMT/UML used in question 6) to implement examples as opposed to code if desired.
7. Explain the difference between runtime binding (polymorphism) and compile time binding (static). Exhibit a minimal example using Java clearly indicating a method invocation requiring a polymorphic bind and a method invocation allowing a compile time bind given the following classes. I am not looking for coding of a complete java system. Just sufficient code to demonstrate the binding mechanisms.

**class pet{**

**// data structures and methods common to all pets**

**public void speak( ){ System.out.println("In Pets");}**

**}**

**class dog extends pet{**

**//void dog( ){ }; // constructors, or use default constructors**

**public void speak() {System.out.println("Bow Wow"); }**

**}**

**class cat extends pet{**

**public void speak( ) {System.out.println("Meow Purr"); }**

**}**

**class bird extends pet{**

**public void speak( ) {System.out.println("Whistle"); }**

**}**

**class fish extends pet{**

**public void speak( ) {System.out.println("Glub, Glub"); }**

**}**

public class JavaPolyMorphism{

public static void main(String[ ] args){

***// Code to answer the question goes here!***