

Black Box Software Testing

Study Guide Questions

Latest Revision: Fall 2010

Copyright (c) Cem Kaner

Notes for Faculty

In my course of software testing, I hand students a study guide at the start of the term. The study guide includes a list of questions.

- In a 1-semester course, my list usually has 50 short-answer and 50 long-answer questions. Other instructors add definitions to their lists (maybe 40 definitions, 40 short-answer and 40 long-answer.)
- In a 4-week Certificate course (focused on one topic), we typically have 25 to 35 questions (mixed short and long answer).

All of my exam questions come from this list. Some other teachers add one or two surprise questions.

Because students have plenty of time to work with these questions, I can expect well-organized, well-focused, thoughtful answers. Along with the list of questions, I provide students with a paper on how to answer exam questions and a long video that demonstrates my approach to grading.

Be thoughtful about the size of the list of study questions that you give students. If the list is too long, students will not try to answer the questions before the exam; they'll just read the answers. The quality of their performance on the final will suffer substantially.

Notes on the Study Guide Questions

- The questions in the study guide often address browser or word processor issues. This is arbitrary. It reflects the applications we've tested recently in class. If your class is testing some other type of application, I suggest that you modify these questions so that most of them reflect that type.
- Many of the questions in this list go beyond the detail that is available in the slides. In the course that I teach, students get several supplementary readings, and I provide additional information in lecture.
- You'll notice that several of the questions are related. This is intentional. Here's an example:
 - **5.L.4.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). List 5 ways that these functions could fail. For each potential type of failure, describe a good test for it, and explain why that is a good test for that type of failure.
 - **5.L.5.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Think in terms of *persistent data*. What persistent data is (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total).
 - **5.L.6.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Think in terms of *data* that you enter into the table. What data is (or could be) associated with tables? List five types of failures that

could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure..

- **5.L.7.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Think in terms of *user interface controls*. What user interface controls are (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total).
- **5.L.8.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Think in terms of *compatibility with other software*. What compatibility features or issues are (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve compatibility. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total).
- **5.L.9.** You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Suppose that a critical requirement for this release is *scalability of the product*. What scalability issues might be present in the table? List three. For each issue, list 2 types of failures that could involve scalability. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total).

These are all about testing the word processor's table functions, but the sequence tells a lesson.

- The first one simply identifies the function and asks the student to imagine potential failures. When students work only on this question, their answers are often superficial and defocused.
- The next three are taken from the Product Elements list of Bach's Heuristic Test Strategy Model.
- The last two are from the Quality Criteria list of Bach's model.
- The six together give the student six different angles on the same problem. The student will come up with different tests, after thinking about different risks. Collectively, they make a point--the more angles that you come at the product from, the more types of failures you can imagine (and the more bugs you can find). These questions thus provide an experience basis for talking about the Strategy model. In Bach's and my experience, students only come to understand the value of this model after a lot of practice.
- You might ask a full sequence of questions like this on an assignment, but you would not (well, *I would not*) ask more than one or two of these questions on an exam. The questions have more value than use on an exam--they guide study, and so you can lead students to insights with a sequence of questions, followed up by comments in lecture after the students have had occasion (e.g. the test) to study.
- It helps to actively encourage students to work together. I host two study sessions before each test or exam. I hope a typical session on a weekend day at the local cafe, and pay for breakfast for anyone who arrives before 1030 a.m. I don't answer the questions for the students. I sit in the cafe, but several tables away, doing my own work. I do help them get back on track when a few of them disagree with each other and can make no progress. (The main benefit of my presence is that it makes clear to the students that I think this is important, and they should come.)
- Grade the first midterm aggressively. You might fail a lot of students. (Things change gradually, as your course develops a reputation.) Don't worry about it. They can rise to reasonable standards. Give comments on the answers, and conduct a review class (or long after-hours review session) in which you walk through questions, point out common mistakes, and give examples of better answers. If you use a detailed outline or checklist for grading, share it with them. And hand out some sample answers.

I prefer good answers written by the students, rather than my own. The samples are imperfect, but they show what can actually be done in the exam situation.

- I often host a makeup midterm, in a 1-semester course.
 - ***This is a voluntary makeup.*** Students who did well on the first midterm won't take it. I schedule the exam outside of class hours, and hold reviews (before and after) out of class hours.
 - ***The makeup replaces the original midterm.*** If the student's grade goes up, good. If the grade goes down, it goes down. This limits the number of exams to grade. You don't get frivolous exams submitted just in case they're OK (which you do get if you use the best of both exams) and you don't get the compulsive A student hoping to raise her grade by 2% . Keeping the number of submissions down is important--this course involves a lot of grading. You need to protect your time.
 - ***I do not always give a makeup midterm.*** I don't promise it in the syllabus and refuse to talk with students about it before they have finished the midterm (except to warn students who ask that there might not be one). This is important--if students know that there is a makeup midterm, some will prepare inadequately for the first midterm with the idea that they can do well on the makeup.
 - These exams are just part of the assessment strategy for the Testing course. They give students a focus for interacting with the lecture and reading materials, but students also need to develop experience with the testing techniques, troubleshooting, and bug reporting. For those, you have to give students assignments and homework, not exams.
-

Fundamentals

Definitions

- Software testing
- Acceptance testing
- Black box testing
- Glass box / white box testing
- Functional testing
- Independent testing
- Parafunctional attributes of a program
- Structural testing
- Behavioral testing
- Unit testing
- Integration testing
- Subsystem testing
- System testing
- Verification
- Validation

Short Answer

S.1. What is the primary difference between black box and glass box testing? What kinds of bugs are you more likely to find with black box testing? With glass box? (If it is not obvious, why would one bug be more easily found by the one approach than the other?)

S.2. Compare and contrast behavioral and black box testing.

S.3. Compare and contrast structural and glass box testing.

S.4. Compare and contrast functional and parafunctional testing.

S.5. Compare and contrast unit versus system testing.

S.6. Compare and contrast verification and validation.

Long Answer

L.1. State, describe, compare and contrast three different definitions of software testing. Which do you prefer? Why?

L2. The lecture summarized the goal of acceptance testing as answering the question, "Should we pay for this program?" What kinds of tests are you *unlikely* to run as part of an acceptance test effort and what kinds of bugs are you more likely to miss as a result? Give three examples of each and explain your reasoning.

Oracles

Definitions

- Heuristic
- Heuristic oracle
- Oracle
- Postcondition data
- Postcondition state
- Precondition data
- Precondition state
- Reference program
- Test oracle (used in automated testing)

Short Answer

S.1. SoftCo makes a word processing program. The program exhibits an interesting behavior. When you save a document that has exactly 32 footnotes, and the total number of characters across all footnotes is 1024, the program deletes the last character in the 32nd footnote.

Think about the "Consistency with History" heuristic. Describe the type of research that you would do, and give an example of the type of argument you could make on the basis of that research, to argue that this behavior is inappropriate.

Note: *Some students abandon their common sense when they answer this question. Think realistically about this bug as you consider your possible answers.*

S.2. Describe three types of oracles used in automated testing.

S.3. In lecture, I asserted that all oracles are heuristic. What is the basis for that assertion? What do you think of that assertion? Why? *Bonus points:* describe a counter-example to this assertion.

S.4. What is a heuristic? Why are heuristics important in software testing?

S.5 What does it mean to specify a test by describing the precondition state of the program (and the system it runs on), the steps you take during the test and the resulting postcondition state? (Define the terms.) Would this be a complete specification of the test? Why or why not? Is it practical to do this? Why or why not?

S.6. What kinds of bugs might you be likely to miss if you rely on automated testing that compares results to a reference program as its oracle? (Unless the answer is obvious, explain briefly why you might be likely to miss these.)

S.7. Describe a situation in which you would use self-verifying data as an oracle. (How would you use it; what type of data would you use it for.)

Long Answer

L.1. The *oracle problem* is the problem of finding a method that lets you determine whether a program passed or failed a test.

Suppose that you were doing automated testing of page layout (such as the display of pages that contained frames or tables) in the Firefox browser.

Describe three different oracles that you could use or create to determine whether layout-related features were working. For each of these oracles,

- identify a bug that would be easy to detect using the oracle. Why would this bug be easy to detect with this oracle? and
- identify another bug that your oracle would be more likely to miss. Why would this bug be harder to detect with this oracle?

L.2. The *oracle problem* is the problem of finding a method that lets you determine whether a program passed or failed a test.

Suppose that you were doing automated testing of page layout (how the spreadsheet or charts based on it will look like when printed) of a Calc spreadsheet. Describe three different oracles that you could use or create to determine whether layout-related features were working. For each of these oracles,

- identify a bug that would be easy to detect using the oracle. Why would this bug be easy to detect with this oracle? and
- identify another bug that your oracle would be more likely to miss. Why would this bug be harder to detect with this oracle?

L.4. The *oracle problem* is the problem of finding a method that lets you determine whether a program passed or failed a test.

Suppose that you were doing automated testing of page layout (how the document will look like when printed) of an HTML editor. Describe three different oracles that you could use or create to determine whether layout-related features were working. For each of these oracles,

- identify a bug that would be easy to detect using the oracle. Why would this bug be easy to detect with this oracle? and
- identify another bug that your oracle would be more likely to miss. Why would this bug be harder to detect with this oracle?

L.3. You can import Microsoft Excel spreadsheets into OpenOffice Calc by opening an Excel-format file with Calc or by copy/pasting sections of the Excel spreadsheet to the Calc spreadsheet. Think about planning the testing of the importation. List 10 types of data or attributes of the data that you should test and for each, briefly describe how you will use an oracle to determine whether the program passed or failed your tests.

L.4. You are using a high-volume random testing strategy for the Mozilla Firefox program. You will evaluate results by using an oracle.

- Consider entering a field into a form, using the entry to look up a record in a database, and then displaying the record's data in the form. How would you create an oracle (or group of oracles)? What would the oracle(s) do?
- Now consider displaying a table whose column widths depend on the width of the column headers. Test this across different fonts (vary typeface and size). How would you create an oracle (or group of oracles) for this? What would the oracle(s) do?
- Which oracle would be more challenging to create or use, and why?

L.5. SoftCo makes a word processing program. The program exhibits an interesting behavior. When you save a document that has exactly 32 footnotes, and the total number of characters across all footnotes is 1024, the program deletes the last character in the 32nd footnote.

- Think about the "Consistency with our Image" heuristic. Describe the type of research that you would do, and give an example of the type of argument you could make on the basis of that research, to argue that this behavior is inappropriate.
- Think about the "Consistency with Comparable Products" heuristic. Describe the type of research that you would do, and give an example of the type of argument you could make on the basis of that research, to argue that this behavior is inappropriate.
- Think about the "Consistency within Product" heuristic. Describe the type of research that you would do, and give an example of the type of argument you could make on the basis of that research, to argue that this behavior is inappropriate.

Note: Some students abandon their common sense when they answer this question. Think realistically about this bug as you consider your possible answers.

L.6. While testing a browser, you find a formatting bug. The browser renders single paragraph blockquotes correctly—it indents them and uses the correct typeface. However, if you include two paragraphs inside the `<blockquote>...</blockquote>` commands, it leaves both of them formatted as normal paragraphs. You have to mark each paragraph individually as blockquote.

Consider the consistency heuristics that we discussed in class. Which three of these look the most promising for building an argument that this is a defect that should be fixed?

For each of the three that you choose:

- Name the heuristic
- Describe research that this heuristic suggests to you
- Describe an argument that you can make on the basis of that research

L.7. Imagine writing a program that allows you to feed commands and data to Microsoft Excel and to Open Office Calc. You have this program complete, and it's working. You have been asked to test a new version of Calc and told to automate all of your testing. What oracles would you use and what types of information would you expect to get from each?

Impossibility of Complete Testing

Definitions

- Complete testing
- Coverage
- Defect arrival rate
- Defect arrival rate curve
- Easter egg
- Line (or statement) coverage
- Side effect (of measurement)
- Surrogate measure

Short Answer

S.1. Consider a program with two loops, controlled by index variables. The first variable increments (by 1 each iteration) from -3 to 20. The second variable increments (by 2 each iteration) from 10 to 20. The program can exit from either loop normally at any value of the loop index. (Ignore the possibility of invalid values of the loop index.)

- If these were the only control structures in the program, how many paths are there through the program?
 - If the loops are nested
 - If the loops are in series, one after the other
- If you could control the values of the index variables, what test cases would you run if you were using a domain testing approach?
- Please explain your answers with enough detail that I can understand how you arrived at the numbers.
- *Note: a question on the test might use different constants but be identical to this question in all other respects.*

S.2. A program asks you to enter a password, and then asks you to enter it again. The program compares the two entries and either accepts the password (if they match) or rejects it (if they don't). An entry is "valid" if it contains only letters and/or digits and is neither too short nor too long.

How many valid entries could you test? (Please show and/or explain your calculations.)

S.3. A program is structured as follows:

- It starts with a loop, the index variable can run from 0 to 20. The program can exit the loop normally at any value of the index.
- Coming out of the loop, there is a case/switch statement that will branch to one of 10 places depending on the value of X. X is a positive, non-zero integer. It has a value from 1 to 10.
- In 9 of the 10 cases, the program executes X statements and then goes into another loop. If X is even, the program can exit the loop normally at any value of its index, from 1 to X. If X is odd, the program goes through the loop 666 times and then exits. In the 10th case (I am explicitly NOT specifying which of the 10 values of X corresponds to the 10th case), the program exits.

Ignore the possibility of invalid values of the index variable or X. How many paths are there through this program? Please show and/or explain your calculations.

Note: a question on the test might use different constants but be identical to this question in all other respects.

S.4. Describe the underlying cause of the MASPAR bug. Compare testing for this bug using black box and glass box approaches.

S.5. Consider the program described by Myers to illustrate calculating number of paths through the program. Change the program as follows: (a) from E, the program can go to H or I or J, not just H or I and (b) Within each iteration of the loop from A to X, the program can loop back from H to A, at most 10 times.

- Draw a chart to show the new program
- Calculate the number of paths through the program.

Note: a test question might use different constants but would be identical to this question in all other respects.

S.6. X is a floating point number, stored to 5 decimal digits of precision. What set of numbers could you input to X that would be stored equivalently to π ? What set of numbers would be equivalent to $\pi/10000$?

Note that π equals
3.141592653589793238462643383279502884197169399375105820...

Note: a test question might use different constants but would be identical to this question in all other respects.

S.7. Let's measure the productivity of programmers by counting their lines (statements) of code. Supposing that this is a measure of performance of programmers, is it a ratio-scaled measure of performance? Why or why not?

Long Answer

L.1. What is the Defect Arrival Rate? Some authors model the defect arrival rate using a Weibull probability distribution. Describe this curve. Suppose that a company measures its project progress using such a curve. Describe and explain the impact of two of the pressures testers are likely to face early in the testing of the product and two of the pressures they are likely to face near the end of the project.

L.2 Consider testing a word processing program, such as Open Office Writer. Describe 5 types of coverage that you *could* measure, and explain a benefit and a potential problem with each. Which one(s) would you actually use and why?

L.3. Define statement coverage, branch coverage, and multicondition coverage. Now describe a small program, describe a set of tests that would completely test it (in terms of statement, branch and multicondition coverage), and describe some bugs that these tests would miss. Your description of the program can be a flowchart or other simplifying diagram.

L.4. Given that complete testing is impossible, what are some potential stopping rules (rules for deciding when to stop testing)? Describe the rules you propose, and for each, explain the tradeoffs involved in using it (what's good about it, what's bad about it, what risks does it introduce, etc.).

L.5. SoftCo publishes software. Their president hates Easter Eggs and has instructed the test group to find every one (if there are any) in the product it is testing. As lead tester, it is your to figure out how to test for

Easter Eggs and when to declare the job done. *How will you decide when you have finished this task? Present your ideas, their strengths and weaknesses.*

L.6. Distinguish between using code coverage to highlight what has not been tested from using code coverage to measure what has been tested. Describe some benefits and some risks of each type of use. (In total, across the two uses, describe three benefits and three risks.)

Domain Testing

Definitions

- Boundary chart
- Boundary condition
- Best representative
- Corner case
- Domain
- Domain testing
- Environment variables
- Equivalence class
- Equivalent tests
- Linearizable variable
- Ordinal variable
- Ordinality
- Ordered domain
- Output domain
- Partition

Short Answer

S.1. Ostrand & Balcer described the category-partition method for designing tests. Their first three steps are:

- Analyze
- Partition, and
- Determine constraints

Describe and explain these steps.

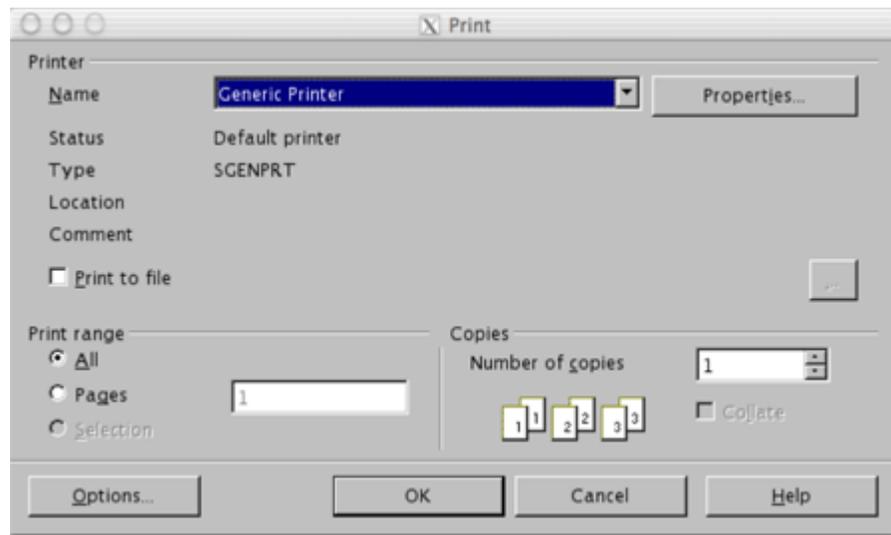
S.2. The overview of domain testing in our workbook includes this step: “Lay out the analysis in a table that shows the partitions and best representatives (or boundary cases) for each partition.”

What do we mean by “partitions” and by “best representatives”?

S.3. Is there ever a difference between a boundary case and a best representative? Explain

S.4. The domain testing workbook differentiates between primary and secondary dimensions. What are the differences? Use examples to clarify your points.

S.5. Here is the print dialog in Open Office.



Suppose that:

1. The largest number of copies you could enter into the Number of Copies field is 999, and
2. Your printer will manage multiple copies (printing the same page repeatedly without reloading it from the connected computer), to a maximum of 99 copies .

For each case, do a traditional domain analysis.

S.3. In the Print Options dialog in Open Office Writer, you can mark (Yes/No) for inclusion on a document:

- Graphics
- Tables
- Drawings
- Controls
- Background

(a) Would you do a domain analysis on these (Yes/No) variables?

(b) What benefit would you gain from such an analysis?

S.4. In the Printer Options dialog in Open Office Impress, you can mark (Yes/No) for inclusion on a document:

- Page name
- Date
- Time
- Hidden pages

(a) Would you do a domain analysis on these (Yes/No) variables? Why or why not?

(b) What benefit(s) (if any) would you gain from such an analysis?

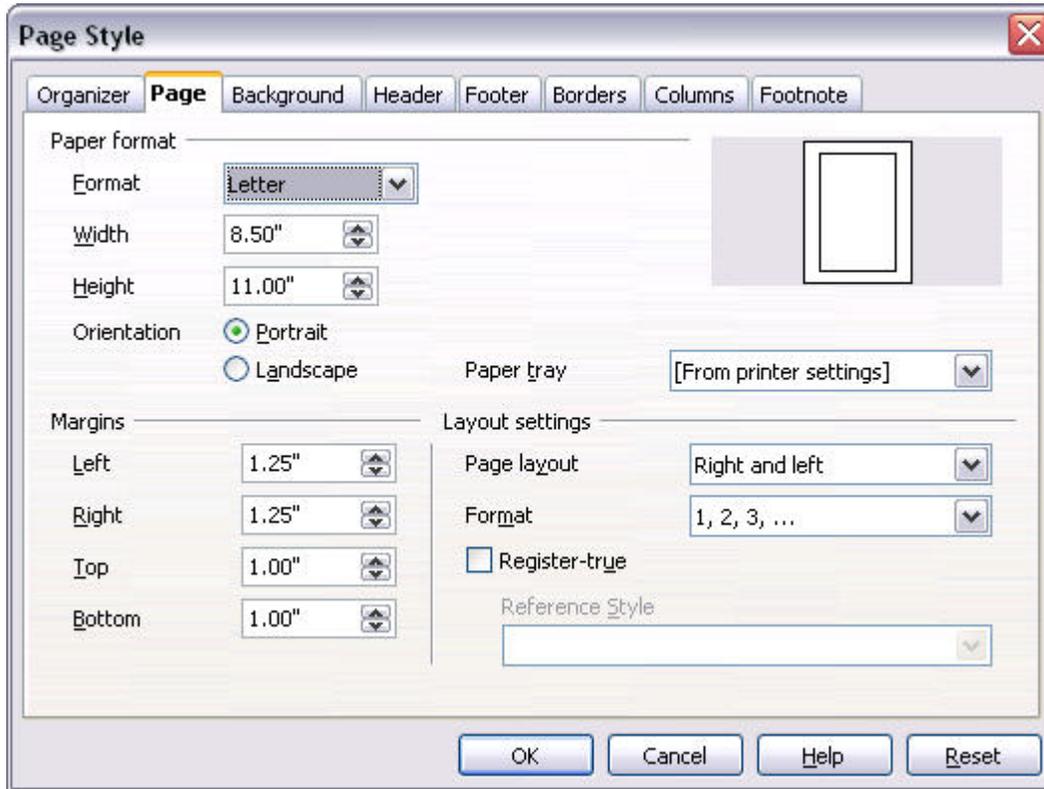
S.5. In the Page Setup dialog in Firefox, you can choose for printing:

- Portrait or landscape orientation
- Whether or not to print background colors and images

(a) Would you do a domain analysis on these (Yes/No) variables? Why or why not?

(b) What benefit(s) (if any) would you gain from such an analysis?

S.6. Here is a Page Style dialog from Open Office



(a) Do a classical boundary and equivalence analysis of **page width**

(b) What's the difference between this and the analysis of an integer?

S.7. How is the notion of a best representative in domain testing different from the notion of a representative value in stratified sampling?

Long Answer

L.1. Imagine testing a date field. The field is of the form MM/DD/YYYY (two digit month, two digit day, 4 digit year). Do an equivalence class analysis and identify the boundary tests that you would run in order to test the field. (Don't bother with non-numeric values for these fields.)

L.2. I, J, and K are signed integers. The program calculates $K = I * J$. For this question, consider only cases in which you enter integer values into I and J. Do an equivalence class analysis **on the variable K** from the point of view of the effects of I and J (jointly) on K. Identify the boundary tests that you would run (the values you would enter into I and J) in your tests of K.

Note: In the exam, I might use $K = I / J$ or $K = I + J$ or $K = \text{IntegerPartOf}(\text{SquareRoot}(I * J))$

L.3. Ostrand & Balcer described the category-partition method for designing tests. Their first three steps are:

1. Analyze
2. Partition, and
3. Determine constraints

Please remember what you read in Ostrand and Balcer's paper. Then apply their method to this function:

I, J, and K are unsigned integers. The program calculates $K = I * J$. For this question, consider only cases in which you enter integer values into I and J.

Do an equivalence class analysis **on the variable K** from the point of view of the effects of I and J (jointly) on K. Identify the boundary tests that you would run (the values you would enter into I and J) in your tests of K.

*Note: In the exam, I might use $K = I / J$ or $K = I + J$ or $K = IntegerPartOf(SquareRoot(I * J))$*

L.4 Our workbook describes several steps in a process for designing domain tests. Here are the first steps:

- A. Identify the potentially interesting variables.
- B. Identify the variable(s) you can analyze now.
- C. Determine the variable's primary dimension.
- D. Determine the variable's type or scale.
- E. Determine whether the variable's values can be ordered (smallest to largest)
- F. Partition (create equivalence classes):
 - a. If the dimension is ordered, determine the sub-ranges and transition points.
 - b. If the dimension is not ordered, determine what "similar" means for this variable, and base partitioning on that.
- G. Lay out the analysis in a table that shows the partitions and best representatives for each partition. Your layout should be the traditional boundary and equivalence class table.
- K. Identify constraints among the variables
- L. Identify and design tests for variables that hold results (output variables).

Apply this method to this function:

- I, J, and K are unsigned integers. The program calculates $K = I * J$. For this question, consider only cases in which you enter integer values into I and J.
- Do an equivalence class analysis **on the variable K** from the point of view of the effects of I and J (jointly) on K. Identify the boundary tests that you would run (the values you would enter into I and J) in your tests of K.

*Note: In the exam, I might use $K = I / J$ or $K = I + J$ or $K = IntegerPartOf(SquareRoot(I * J))$*

L.4. The Spring and Fall changes between Standard and Daylight Savings time creates an interesting problem for telephone bills.

Focus your thinking on the complications arising from the daylight savings time transitions.

Create a table that shows risks, equivalence classes, boundary cases, and expected results for a

long distance telephone service that bills calls at a flat rate of \$0.05 per minute. Assume that the chargeable time of a call begins when the called party answers, and ends when the calling party disconnects.

L.5. Imagine testing a file name field. For example, go to a File Open dialog, you can enter something into the field.

Do a domain testing analysis: List a risk, equivalence classes appropriate to that risk, and best representatives of the equivalence classes.

For each test case (use a best representative), briefly explain why this is a best representative. Keep doing this until you have listed 10 best-representative test cases.

L.6. In EndNote, you can create a database of bibliographic references, which is very useful for writing essays. Here are some notes from the manual:

- Each EndNote reference stores the information required to cite it in a bibliography. Keywords, notes, abstracts, URLs and other information can be stored in a reference as well.
- Each reference added to a library is automatically assigned a unique record number that never changes for that reference in that particular library. . . .
- Each library can reach a record limit of 32,767 or a size of 32MB (whichever comes first). Once a record number is assigned, it cannot be used again in that library. So, if you import 30,000 records, then delete all but 1000 of them, you cannot enter more than another 2767 records into that particular library.

List the variables of interest and do a domain analysis on them.

L.7. Calc allows you to protect cells from modification, and to use a password to override the protection. Think about your testing in terms of *both*, setting the password *and* entering the password later (after it has been set, in order to unprotect a cell).

- How many valid passwords could be created? (Please show and/or explain your calculations.)
- How many invalid passwords could you enter?
- When the program compares the password you enter to the (valid) password already saved, how many test cases are there?
- Do a domain analysis of the initial password entry
- Do a domain analysis of the password validation (entry and comparison with the valid password already saved).

L.8. One of the common ways to describe how to identify variables, their equivalence classes and boundary cases is to rely on the specification. However, we can analyze many variables whether they are well-specified or not. Describe three examples of ordered variables, and explain how we could apply domain testing to each.

L.9. Why is it difficult to apply domain testing to non-ordered sets? Describe a variable whose values would be non-ordered and explain how you would apply a stratified sampling approach to it.

Risk-Based Testing

Bug Taxonomies

Attacks

Operational Profiles

Definitions

- Attack
- Computation constraints
- Failure mode and effects analysis
- Input constraints
- Operational profile
- Output constraints
- Quick test
- Risk-based testing
- Storage constraints

Short Answer

S.1. Describe Bach's heuristic test strategy model and how to apply it.

S.2. What is a quick test? Why do we use them? Give two examples of quick tests.

Long Answer

L.1. Consider testing an HTML form (displayed in Mozilla Firefox) that has you enter data into a table.

- How would you develop a list of risks for this capability? (If you are talking to people, who would you ask and what would you ask them?) (If you are consulting books or records or databases, what are you consulting and what information are you looking for in it?)
- Why is this a good approach for building a list of risks?
- List 10 risks associated with this function.
- For each risk, briefly (very briefly) describe a test that could determine whether there was an actual defect.

L.2. In the Windows version of OpenOffice, you can create a spreadsheet in Calc, then insert it into Writer so that when you edit the spreadsheet file, the changes automatically appear in the spreadsheet object when you reopen the Writer document.

- How would you develop a list of risks for this capability? (If you are talking to people, who would you ask and what would you ask them?) (If you are consulting books or records or databases, what are you consulting and what information are you looking for in it?)
- Why is this a good approach for building a list of risks?
- List 10 risks associated with this function.
- For each risk, briefly (very briefly) describe a test that could determine whether there was an actual defect.

L.3. Imagine that you were testing the Mozilla “Manage bookmarks” feature.

Describe four examples of each of the following types of attacks that you could make on this feature, and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.4. Imagine that you were testing the "Find in this page" feature of the Mozilla Firefox browser. Describe four examples of each of the following types of attacks that you could make on this feature, and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.5. In the Windows version of OpenOffice, you can create a spreadsheet in Calc, then insert it into Writer so that when you edit the spreadsheet file, the changes automatically appear in the spreadsheet object when you reopen the Writer document.

Describe four examples of each of the following types of attacks that you could make on this feature, and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.6. Imagine that you were testing simple database queries with OpenOffice Calc. Describe four examples of each of the following types of attacks that you could make on this feature, and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.7. Imagine testing the creation and use of templates in Open Office Impress.

Describe four examples of each of the following types of attacks that you could make on this feature, and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.8. Imagine testing a file name field. For example, go to a File Open dialog, you can enter something into the File Name field.

Describe four examples of each of the following types of attacks that you could make on this feature (File Name in this dialog), and for each one, explain why your example is a good attack of that kind.

- Input-related attacks
- Output-related attacks
- Storage-related attacks
- Computation-related attacks.

(Refer specifically to Whittaker, How to Break Software and use the types of attacks defined in that book. Don't give me two examples of what is essentially the same attack. In the exam, I will not ask for all 16 examples, but I might ask for 4 examples of one type or two examples of two types, etc.)

L.9. You are testing the group of functions that let you format a table in Open Office Calc.

List 5 ways that these functions could fail. For each potential type of failure, describe a good test for it, and explain why that is a good test for that type of failure. (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.10. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

List 5 ways that these functions could fail. For each potential type of failure, describe a good test for it, and explain why that is a good test for that type of failure. (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there,

explain what about the test justifies your assertion that the test is powerful.)

L.11. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Think in terms of *persistent data*. What persistent data is (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.12. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Think in terms of *data* that you enter into the table . What data is (or could be) associated with tables? List five types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.13. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Think in terms of *user interface controls* . What user interface controls are (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.14. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Think in terms of *compatibility with external software*. What compatibility features or issues are (or could be) associated with tables? List three types. For each type, list 2 types of failures that could involve compatibility. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.15. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Suppose that a critical requirement for this release is **scalability of the product**. What scalability issues might be present in the table? List three. For each issue, list 2 types of failures that could involve scalability. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you

explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.16. You are testing the group of functions that let you create and format a spreadsheet.

Think in terms of *persistent data* (other than the data you enter into the cells of the spreadsheet). What persistent data is (or could be) associated with a spreadsheet? List three types. For each type, list 2 types of failures that could involve that data. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.17. You are testing the group of functions that let you create and format a spreadsheet.

Think in terms of *compatibility with external software*. What compatibility features or issues are (or could be) associated with spreadsheets? List three types. For each type, list 2 types of failures that could involve compatibility. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

L.18. You are testing the group of functions that let you create and format a spreadsheet..

Suppose that a critical requirement for this release is **scalability of the product**. What scalability issues might be present in a spreadsheet? List three. For each issue, list 2 types of failures that could involve scalability. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

Scenario Testing

Definitions

- Scenario
- Soap opera
- Credibility of a test

Short Answer

S.1. Describe the characteristics of a good scenario test.

S.2. Compare and contrast scenario testing and beta testing. (Give one comparison and one contrast)

S.3. Compare and contrast scenario testing and specification-based testing.(Give one comparison and one

contrast)

S.4. Why would you use scenario testing instead of domain testing? Why would you use domain testing instead of scenario testing?

Long Answer

L.1. Define a scenario test and describe the characteristics of a good scenario test.

Imagine developing a set of scenario tests for handling security certificates in Mozilla Firefox.

- What research would you do in order to develop a series of scenario tests?
- Describe two scenario tests that you would use and explain how these would relate to your research and why each is a good test.

L.2. Imagine that you were testing how Mozilla Firefox's Password manager saves login passwords.

- Explain how you would develop a set of scenario tests that test this feature.
- Describe a scenario test that you would use to test this feature.
- Explain why this is a particularly good scenario test.

L.3. Imagine that you were testing how Mozilla Firefox's Password manager saves login passwords.

- Explain how you would develop a set of soap operas that test this feature.
- Describe one test that might qualify as a soap opera.
- Explain why this is a good soap opera test.

L.4. Define a scenario test and describe the characteristics of a good scenario test.

Imagine developing a set of scenario tests for charting in Calc?

- What research would you do in order to develop a series of scenario tests?
- Describe two scenario tests that you would use and explain how these would relate to your research and why each is a good test.

L.5. Imagine that you were testing how Calc protects cells from modification.

- Explain how you would develop a set of scenario tests that focus on this feature.
- Describe a scenario test that you would use while testing this feature.
- Explain why this is a particularly good scenario test.

L.6. Imagine that you were testing how Calc protects cells from modification.

- Explain how you would develop a set of soap operas that focus on this feature.
- Describe one test that might qualify as a soap opera.
- Explain why this is a particularly good soap opera test.

L.7. Suppose that scenario testing is your primary approach to testing. What controls would you put into place to ensure good coverage? Describe at least three and explain why each is useful.

L.8. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office). Think about the different types of users of word processors. Why would

they want to create tables? Describe three different types of users, and two types of tables that each one would want to create. (In total, there are 3 users, 6 tables). Describe a scenario test for one of these tables and explain why it is a good scenario test.

L.9. You're testing the Firefox browser. The general area that you're testing is handling of tables. What do people do with tables in browsers? Give five examples, one each of five substantially different uses of tables. (In total, there are 5 examples.) Now consider our list of 12 ways to create good scenarios, and focus on "Try converting real-life data from a competing or predecessor application." Describe two scenario tests based on these considerations. For one of them, explain why it is a good scenario test.

L.10. You're testing the goal-seeking function of Calc. What do people do with goal-seeking? Give three examples, one each of three substantially different uses of goal-seeking. (In total, there are 3 examples.) Now consider our list of ways to create good scenarios, and focus on "List possible users, analyze their interests and objectives." Describe two scenario tests based on these considerations. For one of them, explain why it is a good scenario test.

L.11. Define a scenario test and describe the characteristics of a good scenario test.

Imagine developing a set of scenario tests for AutoCorrect in OpenOffice Writer.

- What research would you do in order to develop a series of scenario tests?
- Describe two scenario tests that you would use and explain how these would relate to your research and why each is a good test.

L.12. Imagine that you were testing how OpenOffice Writer does outline numbering.

- Explain how you would develop a set of scenario tests that focus on this feature.
- Describe a scenario test that you would use while testing this feature.
- Explain why this is a particularly good scenario test.

L. 13 Imagine that you were testing how Open Office formats tables in Impress.

- Explain how you would develop a set of scenario tests that focus on this feature.
- Describe a scenario test that you would use while testing this feature.
- Explain why this is a particularly good scenario test. (IMPORTANT: Refer back to the characteristics of good scenario tests)

Test Design

Definitions

- Power of a test
- Validity of a test
- Credibility of a test
- Motivational value of a test
- Complexity of a test
- Opportunity cost of a test
- Test idea
- Test procedure
- Information objectives
- Test technique
- Test case

Short Answer

- S.1. List and describe four different dimensions (different “goodnesses”) of “goodness of tests”.
- S.2. Give three different definitions of a test case. Which is the best one (in your opinion) and why?
- S.3. What is opportunity cost and why is it such an important issue in testing?
- S.4. What is the power of a test? Credibility of a test? Contrast them with an example of a good test that has high power/low credibility and another that has low power/high credibility.
- S.5. Suppose you were testing a spreadsheet. Consider testing with each of these two information objectives: “Assess conformance to specifications” versus “Block premature product releases.” How might your testing be similar and how might it be different under these objectives.

Long Answer

L.1. Suppose that a test group's mission is to achieve its primary information objective. Consider (and list) three different objectives. For each one, how would you focus your testing? How would your testing differ from objective to objective?

L.2. The course notes describe a test technique as a recipe for performing the following tasks:

- Analyze the situation
- Model the test space
- Select what to cover
- Determine test oracles
- Configure the test system
- Operate the test system
- Observe the test system
- Evaluate the test results

Compare and contrast how *scenario testing* and *regression testing* guide us in performing each of these tasks?

L.3. The course notes describe a test technique as a recipe for performing the following tasks:

- Analyze the situation
- Model the test space
- Select what to cover
- Determine test oracles
- Configure the test system
- Operate the test system
- Observe the test system
- Evaluate the test results

- Compare and contrast how *scenario testing* and *domain testing* guide us in performing each of these tasks?

L.4. The course notes describe a test technique as a recipe for performing the following tasks:

- Analyze the situation

- Model the test space
- Select what to cover
- Determine test oracles
- Configure the test system
- Operate the test system
- Observe the test system
- Evaluate the test results

How does *specification-based testing* guide us in performing each of these tasks?

L.5. The course notes describe a test technique as a recipe for performing the following tasks:

- Analyze the situation
- Model the test space
- Select what to cover
- Determine test oracles
- Configure the test system
- Operate the test system
- Observe the test system
- Evaluate the test results

How does *risk-based testing* guide us in performing each of these tasks?

L.6. Consider *domain testing* and *specification-based testing*. What kinds of bugs are you more likely to find with domain testing than with specification-based testing? What kinds of bugs are you more likely to find with specification-based testing than with domain testing?

L.7. Consider *scenario testing* and *function testing*. What kinds of bugs are you more likely to find with *scenario* testing than with *function* testing? What kinds of bugs are you more likely to find with *function* testing than with *scenario* testing?

Function Testing

Definitions

- Function testing

Short Answer

S.1. When would you use function testing and what types of bugs would you expect to find with this style of testing?

S.2. Advocates of GUI-level regression test automation often recommend creating a large set of function tests. What are they actually advocating and why? What are some benefits and risks of this?

S.3. What is a function list and how would you build one for Open Office Impress?

Long Answer

Matrices

Definitions

- Test matrix

Short Answer

S.1. Describe two benefits and two risks associated with using test matrices to drive your more repetitive tests.

Long Answer

Specification-Based Testing

Definitions

- Active reading
- Implicit specifications
- Specification-based testing
- Testability
- Traceability matrix

Short Answer

S.1. What kinds of errors are you likely to miss with specification-based testing?

S.2. What is an implicit specification? Give an example and describe how it is used in test design or bug reporting.

Long Answer

L.1. Describe a traceability matrix.

- How would you build a traceability matrix for the display rules in Mozilla Firefox?
- What specification(s) are you tracing to and where would you get them?
- What is the traceability matrix used for?
- What are the advantages and risks associated with driving your testing using a traceability matrix?
- Give examples of advantages and risks that you would expect to deal with if you used a traceability matrix for the display rules. Answer this in terms of two of the display rules that you can change in Mozilla Firefox. You can choose which two features.

L.2. Describe a traceability matrix.

- How would you build a traceability matrix for the database access features in Calc?
- What specification(s) are you tracing to and where would you get them?
- What is the traceability matrix used for?
- What are the advantages and risks associated with driving your testing using a traceability matrix?
- Give examples of advantages and risks that you would expect to deal with if you used a traceability

matrix for the display rules. Answer this in terms of two of the database access features that you can use in Calc. You can choose which two features.

L.3. Describe a traceability matrix.

- How would you build a traceability matrix for Open Office's word processor?
- What specification(s) are you tracing to and where would you get them?
- What is the traceability matrix used for?
- What are the advantages and risks associated with driving your testing using a traceability matrix?
- Give examples of advantages and risks that you would expect to deal with if you used a traceability matrix for any two of the following features of Open Office Writer:
 - Outlines
 - Tables
 - Fonts
 - Printing

L.4 What is active reading? Explain how these study guide questions are a tool for helping you actively read/process the course videos and slides? Describe two other active reading techniques that you could use with the course videos and slides, and how you would use them?

L.17. Describe a traceability matrix.

- How would you build a traceability matrix for the Tables feature in Open Office's word processor?
- What specification(s) are you tracing to and where would you get them?
- What is the traceability matrix used for?
- What are the advantages and risks associated with driving your testing using a traceability matrix?
- Give examples of advantages and risks that you would expect to deal with if you used a traceability matrix for Tables. Answer this in terms of two of the Tables features that you can change in OpenOffice Writer. You can choose which two features.

L.17. Describe a traceability matrix.

- How would you build a traceability matrix for the Tables feature in Open Office's presentation program?
- What specification(s) are you tracing to and where would you get them?
- What is the traceability matrix used for?
- What are the advantages and risks associated with driving your testing using a traceability matrix?
- Give examples of advantages and risks that you would expect to deal with if you used a traceability matrix for Tables. Answer this in terms of two of the Tables features that you can change in OpenOffice Impress. You can choose which two features.

Regression Testing

Definitions

- Bug regression
- Change detector
- General functional regression
- Old-fix regression
- Regression testing

- Smoke testing

Short Answer

S.1. What risks are we trying to mitigate with black box regression testing?

S.2. What risks are we trying to mitigate with unit-level regression testing?

S.3. What are the differences between risk-oriented regression testing and procedural regression testing?

Long Answer

L.1. What is regression testing? What are some benefits and some risks associated with regression testing? Under what circumstances would you use regression tests?

L.2. In lecture, I used a minefield analogy to argue that variable tests are better than repeated tests. Provide five counter-examples, contexts in which we are at least as well off reusing the same old tests.

User Testing

Definitions

- Beta testing
- Usability testing
- User interface testing
- User testing

Short Answer

S.1. How would you organize a beta test with the objective of configuration testing? Why?

S.2. How would you organize a beta test with the objective of assessing the usefulness of the product?

Long Answer

L.1. The traditional beta test starts after most of the code has been written. List and describe three benefits and three risks of starting beta testing this late.

Stress Testing

Definitions

- Buffer overflow
- Load testing
- Stress testing

Short Answer

S.1. What's the difference between stress testing and input testing with extreme values?

S.2. What's the difference between stress testing and load testing?

Long Answer

Stochastic Testing

State Models

Definitions

- Directed graph
- Dumb monkey
- Finite state machine
- Markov chain
- Random testing
- Smart monkey
- State
- State explosion
- State variable
- Statistical reliability estimation
- Stochastic testing
- Value of a state variable

Short Answer

S.1. Describe two difficulties and two advantages of state-machine-model based testing.

S.2. Can you represent a state machine graphically? If so, how? If not, why not?

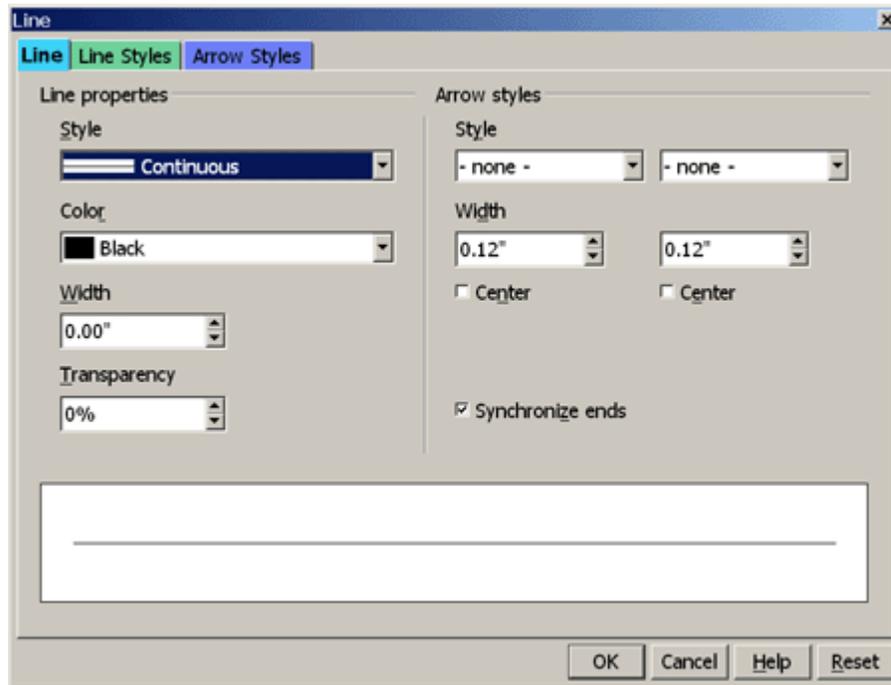
S.3. Explain the relationship between graph traversal and our ability to automate state-model-based tests.

S.4. Compare and contrast the adjacency and incidence matrices. Why would you use one instead of the other?

S.5. What does it tell us about the system under test if the model of system (accurately) shows weak connectivity?

S.6. What is the state explosion problem and what are some of the ways that state-model-based test designers use to cope with this problem?

S.7. Consider this dialog from Open Office Presentation



In this dialog, the variable "Synchronize ends" can be checked or unchecked. Are these two values distinct? Justify your answer.

S.8. What is the difference between "random testing" and "stochastic testing"?

Long Answer

L.1. Suppose you use a state model to create a long random series of tests. What should you use as a stopping rule? Compare three alternatives, one of them being that you stop after the program runs without error for a sequence of 200 computer-hours. What assurance do you have of sub-sequence coverage in these cases?

L.2. Schroeder and Bach argue that there is little practical difference between all-pairs coverage and an equivalent random sample of combination tests. What do you think? From the list of attributes of good tests, list three that might distinguish between these approaches and explain why.

GUI Automation

Definitions

- Capture-replay
- Comparison oracle
- Inertia
- Performance benchmarking
- Smoke testing

Short Answer

S.1. Describe three factors that influence automated test maintenance cost.

S.2. Describe three risks of capture-replay automation.

- S.3. Why do we argue that even capture-replay automation is software engineering?
- S.4. Under what circumstances might capture-replay automation be effective?
- S.5. Should you rerun all (or most) of your automated tests in every build? Why or why not?
- S.6. Describe three of the costs of GUI-level regression automation.
- S.7. What are the benefits and problems of screen capture and comparison?
- S.8. What kinds of bugs might you be likely to miss if you use a reference program as a comparison oracle?

Long Answer

- L.1. Why is it important to design maintainability into automated regression tests? Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.
- L.2. Are GUI regression tests necessarily low power? Why do you think so? What could be done to improve testing power? (NOTE: This is a *long answer* question.)
- L.3. Why does it take 3-10 times as long to create an automated regression test as to create and run the test by hand? How could you improve this ratio? What factors push this ratio even higher?
- L.4. A client retains you as a consultant to help them introduce GUI-level test automation into their processes. What questions would you ask them (up to 7) and how would the answers help you formulate recommendations?
- L.5. A client retains you as a consultant to help them use a new GUI-level test automation tool that they have bought. They have no programmers in the test group and don't want to hire any. They want to know from you what are the most effective ways that they can use the tool. Make and justify three recommendations (other than "hire programmers to write your automation code" and "don't use this tool"). In your justification, list some of the questions you would have asked to develop those recommendations and the type of answers that would have led you to those recommendations.
- L.6. Why do we say that GUI-level regression testing is computer-assisted testing, rather than full test automation? What would you have to add to GUI-level regression to achieve (or almost achieve) full automated testing? How much of this could a company actually achieve? How?
- L.7. Contrast developing a GUI-level regression strategy for a computer game that will ship in one release (there won't be a 2.0 version) versus an in-house financial application that is expected to be enhanced many times over a ten-year period.
- L.8. How should you document your GUI-level regression tests? Pay attention to the costs and benefits of your proposals.

High Volume Automated Testing

Architectures of Test Automation

Definitions

- Comparison function

- Computational oracle
- Delayed-effect bug
- Deterministic oracles
- Diagnostics-based stochastic testing
- Evaluation function
- Extended random regression
- Framework for test automation
- Function equivalence testing
- Heuristic oracles
- Hostile data streams
- Instrumenting a program
- Inverse oracle
- Mutation testing
- Probabilistic oracles
- Probes
- Software simulator
- Wild pointer
- Wrapper

Short Answer

S.1. Describe three risks of focusing your testing primarily on high-volume test techniques.

S.2. How does extended random regression work? What kinds of bugs is it good for finding?

S.3. Describe three practical problems in implementing extended random regression.

S.4. How can it be that you don't increase coverage when using extended random regression testing but you still find bugs?

S.5. Why should load testing expose functional errors?

Long Answer

L.1. Suppose that you had access to the Mozilla Firefox source code and the time / opportunity to revise it. Suppose that you decided to use a *diagnostics-based* high volume automated test strategy to test Firefox's treatment of links to different types of files.

- What diagnostics would you add to the code, and why?
- Describe 3 potential defects, *defects that you could reasonably imagine would be in the software that handles downloading or display of linked files*, that would be easier to find using a diagnostics-based strategy than by using a lower-volume strategy such as exploratory testing, spec-based testing, or domain testing.

L.2. Go back to the Weibull model for defect arrival rate. Which of its assumptions are challenged by our results with extended random regression testing? Explain.

L.3. The simplest form of hostile data stream testing randomly mutates data items. Are there any AI algorithms that could make this technique more efficient? Describe some approaches that you might try.

L.4. Doug Hoffman's description of the square root bug in the MASPAC computer provides a classic example of function equivalence testing. What did he do in this testing, why did he do it, and what strengths

and challenges does it highlight about function equivalence testing?

L.5. This question assumes that you have done an assignment in which you instrumented a program that you wrote with probes.

Describe your results with the probes. What kinds of bugs did you find? What kinds do you think you missed? What practical challenges would you expect to have with this technique on a real project?

L.6. Think about the personnel / staffing issues associated with high-volume test automation. Characterize three of the high-volume techniques in terms of the skills required of the staff. If you were managing a typical testing group, which has few programmers, which technique would you start with and why?

L.7. How should you document automated tests created in a high-volume automated testing context? Pay attention to the costs and benefits of your proposals and to the suitability of your documentation for outside auditors.

L.8. On a Windows system, what are some of the diagnostics that you can build into a diagnostics-based test automation system? How would this work? What kinds of bugs would these help you find?

L.9. Describe two types of oracle and compare the challenges in implementing and using each.

Exploratory Testing

Definitions

- Active reading
- Ambiguity
- Automatic logging tool
- Charter (of a testing session)
- Exploratory fork
- Exploratory testing
- Guideword heuristic
- Heuristic
- Heuristic procedure
- Inattentional blindness
- Lightweight process
- Session
- Software testing
- Subtitle heuristic
- Time box
- Trigger heuristic

Short Answer

S.1. What is a trigger heuristic? Describe and explain how to use an example of a trigger heuristic *other than "no questions."*

S.2. What is a guideword heuristic? Describe and explain how to use an example of a guideword heuristic other than *"buggy."*

S.3. What is a subtitle heuristic? Describe and explain how to use an example of a subtitle heuristic other

than *"no one would do that."*

S.4. What is a heuristic model? Describe and explain how to use an example of a heuristic other than the *"test strategy model"*

S.5. What is a heuristic procedure? Describe an example of a heuristic procedure, and explain why you call it a heuristic.

S.6. Describe three risks of exploratory testing.

Long Answer

L.1. Compare exploratory and scripted testing. What advantages (name three) does exploration have over creating and following scripts? What advantages (name three) does creating and following scripts have over exploration?

L.2. Look up a discussion of the concept of "Guidewords" in HAZOP. How does this idea apply to Bach's Heuristic Test Strategy Model? How would you use this concept and HSTM to guide exploration?

Paired Exploratory Testing

Definitions

- Paired programming

Short Answer

S.1. What makes paired testing effective as a vehicle for training testers? What would make it less effective?

Long Answer

L.1. A company with a large IT department retains you as a consultant. After watching the testers work and talking with the rest of the development staff, you recommend that the testers work in pairs. An executive challenges you, saying that this looks like you're setting two people to do one person's work. How do you respond? What are some of the benefits of paired testing? Are there problems in realizing those benefits? What?

Multi-Variable Testing

Definitions

- All-pairs combination testing
- All-singles combination testing
- All-triples combination testing
- Combination testing
- Combinatorial combination testing
- Data relationship table
- Strong combination testing
- Weak combination testing

Short Answer

S.1. What is a combination chart? Draw one and explain its elements.

S.2. What is strong combination testing? What is the primary strength of this type of testing? What are two of the main problems with doing this type of testing? What would you do to improve it?

S.3. What is weak combination testing? What is the primary strength of this type of testing? What are two of the main problems with doing this type of testing? What would you do to improve it?

S.4. What is the difference between all-pairs and all-triples testing?

Long Answer

L.1. We are going to do some configuration testing on the Mozilla Firefox Office browser. We want to test it on

- Windows 98, 2000, XP home, and XP Pro (the latest service pack level of each)
- Printing to an HP inkjet, a LexMark inkjet, an HP laser printer and a Xerox laser printer
- Connected to the web with a dial-up modem (28k), a DSL modem, a cable modem, and a wireless card (802.11b)
- With a 640x480 display, an 800 x 600 display, a 1024x768 display and a an 1152 x 720 display
 - How many combinations are there of these variables?
 - Explain what an all-pairs combinations table is
 - Create an all-pairs combinations table. (Show at least some of your work.)
 - Explain why you think this table is correct.

Note: In the exam, I might change the number of operating systems, printers, modem types, or display

L.2. We are going to do some configuration testing on the Windows version of Open Office. We want to test it on

- Windows 2000, XP, Vista, and Windows 7 (the latest service pack level of each)
- Printing to a Kodak inkjet, a LexMark inkjet, an HP laser printer and a Xerox laser printer
- Connected to the web with a dial-up modem (28k), a DSL modem, a cable modem, and a wireless card (802.11n)
- Using a mouse, a trackball, a touchpad, or the keyboard
- With a 640x480 display, an 800 x 600 display, a 1024x768 display and a an 1152 x 720 display
 - How many combinations are there of these variables?
 - Explain what an all-pairs combinations table is
 - Create an all-pairs combinations table.
 - Explain why you think this table is correct.

Note: In the exam, I might change the number of operating systems, printers, modem types, input devices, or displays

L.3. We are going to do some configuration testing on the Windows version of Open Office. We want to test it on

- Windows XP, Vista, and Windows 7 (the latest service pack level of each)
- Printing to an inkjet, an HP laser printer and a Xerox laser printer
- Connected to the web with a DSL modem, a cable modem, and a wireless card (802.11n)

- Using a mouse, a touchpad, and the keyboard
- With a 1024x768 display, an 1152 x 720, and a 1600 x 1200 display
- With 1 gB of Ram, 2 gB and 4 gB of RAM
 - How many combinations are there of these variables?
 - Explain what an all-pairs combinations table is
 - Create an all-pairs combinations table.
 - Explain why you think this table is correct.

Note: In the exam, I might change the number of operating systems, printers, modem types, input devices, memory or displays

L.2. Compare and contrast all-pairs testing and scenario testing. Why would you use one over the other?

L.3. What is a data relationship table? Draw one and explain its elements. When would you use a table like this and what would you use it for?

L.4. We considered all-pairs as an approach for strictly independent variables. Why is this restriction important? Suppose you were testing four variables (A, B, C, D) that had 3 values each (A1, A2, A3 and similarly for B, C and D) but A1 was always an invalid combination with B2. How would you modify your all-pairs test set to handle this one dependence? Now, suppose that A1 is ALSO invalid with C3 or D3 and A2 is always invalid with B3, C2 or D3, and C1 is invalid with D2. How would you design your combination tests in this case?

L.5. Compare and contrast the mechanical, risk-based, and scenario-based approaches to developing combination tests. Discuss their strengths and weaknesses relative to each other. Use examples to clarify your points.

Questioning Skills

Definitions

- Behavioral questions
- Closed questions
- Context-dependent questions
- Context-free questions
- Factual questions
- Hypothetical questions
- Open-ended questions
- Opinion-eliciting questions
- Predictive questions

Short Answer

S.1. What are the reporters' questions? Why do we call them context-free?

S.2. Why is it useful to have a collection of context-free questions? What are context-free questions? How would you use them?

S.3. What's the difference between a process question and a product question?

Long Answer

Test Documentation (Introduction)

Definitions

- Architecture diagram
- Combination test table
- Configuration planning table
- Configuration test matrix
- Configuration variable
- Decision table
- Function list
- List (in a set of test documentation)
- Model
- Objectives outline
- Outline (in a set of test documentation)
- Platform variable
- Protocol specification
- State chart
- SUT
- Table (in a set of test documentation)
- Test case
- Test documentation set
- Test plan
- Testing project plan
- Test matrix
- Test suite

Short Answer

S.1. What is a configuration test matrix? Draw one and explain its elements.

S.2. What is a decision table? Draw one and explain its elements.

S.3. What is a combination test table? Draw one and explain its elements.

S.4. What is a decision table? Draw one and explain its elements.

S.5. What is a test matrix? Draw one and explain its elements.

S.6. What is a state chart? Draw one and explain its elements.

S.7. What is a function list? Give a short example of one and explain its elements.

Long Answer

L.1. What's a testing project plan? Describe some of the elements of a testing project plan. What are some of the costs and benefits of creating such a document?

Scripted Manual Test Cases

Definitions

- Inattentional blindness
- Manual test script
- Procedural testing

Short Answer

S.1. Distinguish between a test script and a task checklist.

S.2. Suppose that your company decided to script several hundred tests. What types of tests would you write scripts for? Why?

S.37. What is inattentional blindness and how is it relevant to testing?

Long Answer

L.1. List and explain four claimed strengths of manual scripted tests and four claimed weaknesses.

L.2. Your company decides to outsource test execution. Your senior engineers will write detailed test scripts and the outside test lab's staff will follow the instructions. How well do you expect this to work? Why?

L.3. Suppose your company decides to write test scripts in order to foster repeatability of the test across testers. What are the best arguments in favor of this? What are the best arguments against it?

L.4. Why would you script a test? What aspects of a test would lead you to consider it a reasonable candidate for scripting (if you were required to do some scripting), and why? What aspects of a test would cause you to consider it less suitable for scripting and why?

Requirements for Test Documentation

Definitions

- Disfavored stakeholder
- Favored stakeholder
- IEEE Standard 829
- Project inertia
- Stakeholder
- Interest of a stakeholder
- Traceability of test documentation

Short Answer

S.1. What do you think is a reasonable ratio of time spent documenting tests to time spent executing tests? Why?

- S.2. How would you go about determining what would be a reasonable ratio of time spent documenting tests to time spent executing tests for a particular company or project?
- S.3. How long should it take to document a test case? What can you get written in that amount of time?
- S.4. What does it mean to do maintenance on test documentation? What types of things are needed and why?
- S.5. Why would a company start its project by following Standard 829 but then abandon that level of test documentation halfway through testing?
- S.6. What factors drive up the cost of maintenance of test documentation?
- S.7. How would you document your tests if you are doing high volume automated testing?
- S.8. Does detailed test documentation discourage exploratory testing? If so, how? Why?
- S.9. How can test documentation support delegation of work to new testers? What would help experienced testers who are new to the project? What would help novice testers?
- S.10. How could a test suite support *prevention* of defects?
- S.11. How could test documentation support *tracking* of project status or testing progress?
- S.12. Suppose that your testing goal is to demonstrate *nonconformance with customer expectations*. If you were designing test documentation, how would you design it to support that goal? How do those design decisions support those goals?

Long Answer

L.1. Imagine that you are an external test lab, and Mozilla.org comes to you with Firefox. They are considering paying for some testing, but before making a commitment, they want to understand your approach and get a sense of what they'll get and how much it will cost. So, they ask you what test documentation they (or you) should create. You ask what they want, and they say that they want to rely on your expertise.

How will you decide what test documentation to give them?

To decide what to give them, what questions would you ask (up to 7 questions) and for each answer, how would the answer to that question guide you?

L.2. Imagine that you are an external test lab, and Sun came to you to discuss testing of Open Office Calc. They are considering paying for some testing, but before making a commitment, they need to know what they'll get and how much it will cost.

How will you decide what test documentation to give them?

(Suppose that when you ask them what test documentation they want, they say that they want something appropriate but they are relying on your expertise.)

To decide what to give them, what questions would you ask (up to 7 questions) and for each answer, how would the answer to that question guide you?

L.3. Consider the University's database system that tracks your grades and prints your grade reports at the end of the term.

- List two stakeholders of this system (Identify them by function, not by personal name. You don't have to know the exact name or job title of the actual people. This is a hypothetical example.)
- For each stakeholder, list and briefly describe three interests that you think this stakeholder probably has. Are these favored or disfavored interests?
- For each interest (there are a total of six), list and briefly describe an action that the system can perform that would support the interest.

L.4. Consider the Open Office word processor and its ability to read and write files of various formats.

- List two stakeholders for whom this type of feature would be important. (Identify them by function, not by personal name. You don't have to know the exact name or job title of the actual people. This is a hypothetical example.)
- For each stakeholder, list and briefly describe three interests that you think this stakeholder probably has. Are these favored or disfavored interests?
- For each interest (there are a total of six), how does file handling relate to that interest?

L.5. Consider a browser and its ability to display pages that contain an embedded video (and call an appropriate player to play the video).

- List two stakeholders for whom this type of feature would be important. (Identify them by function, not by personal name. You don't have to know the exact name or job title of the actual people. This is a hypothetical example.)
- For each stakeholder, list and briefly describe three interests that you think this stakeholder probably has. Are these favored or disfavored interests?
- For each interest (there are a total of six), how does this feature relate to that interest?

L.6. Suppose that Boeing developed a type of fighter jet and a simulator to train pilots to fly it. Suppose that Electronic Arts is developing a simulator game that lets players "fly" this jet. Compare and contrast the test documentation requirements you would consider appropriate for developers of the two different simulators.

Test Planning

Definitions

- complexity of a test
- critical path
- dependencies (in a schedule)
- implied requirements
- meticulous testing
- private bugs
- public bugs
- sympathetic testing
- test plan
- test strategy
- test logistics

Short Answer

S.1. What benefits do you expect from a test plan? Are there circumstances under which these benefits would

not justify the investment in developing the plan?

S.2. Describe the characteristics of a good test strategy.

S.3. What do we mean by "diverse half-measures"? Give some examples.

S.4. Explain the statement, "Test cases and procedures should manifest the test strategy." Use an example in your explanation.

S.5. Discuss the assertion that a programmer shouldn't test her own code. Replace this with a more reasonable assertion and explain why it is more reasonable.

S.6. How could you design tests of implicit requirements? Give some examples that illustrate your reasoning or approach.

S.7. Why is it important for test documentation to be as concise and nonredundant as possible?

S.8. Late in a project, after you have been testing for a long time, you find a serious bug that will require expensive revisions to an internal data structure. The project manager is furious because this type of change would have been much cheaper if the bug had been found much earlier. You both check, and determine that the bug was in the code before you ever started testing--you've missed it all these months. How would you decide whether missing this bug up to now was the result of inadequate testing, an erroneous testing strategy, or sensible conformance to a reasonable test strategy?

Long Answer

L.1. Describe four characteristics of a good test strategy. Describe a specific testing strategy for Open Office, and explain (in terms of those four characteristics) why this is a good strategy.

L.2. In the slides, we give the advice, "Over time, the objectives of testing should change. Test sympathetically, then aggressively, then increase complexity, then test meticulously." Explain this advice. Why is it (usually) good advice. Give a few examples, to apply it to the testing of Open Office's presentation program. Are there any circumstances under which this would be poor advice?

L.3. Some testers prefer to build their test documentation around test templates. What are some of the benefits and some of the risks of templates?

L.4. What is the "critical path" of a project? Why is testing often on the critical path? Is it a problem for testing to be on the critical path? (If so, why? If not, why not?) What would increase the extent to which testing is on the critical path? What would decrease that extent?

Bugs and Errors and Quality

Definitions

- Bug
- Coding error
- Customer satisfier
- Defect
- Design error
- Dissatisfier
- Documentation error

- Fault vs. failure vs. defect
- Feedback loop between testers and other developers
- Parafunctional attributes
- Public bugs vs. private bugs
- Quality
- Requirements error
- Software quality
- Specification error

Short Answer

S.1. What is Weinberg's definition of software quality? If your group works with this definition of quality, what is the most appropriate definition of a bug? Why?

S.2. Use Weinberg's definition of quality. Suppose that the software behaves in a way that you don't consider appropriate. Does it matter whether the behavior conflicts with the specification? Why? **And** why not? (explain both)

S.3. Use Crosby's definition of quality. Suppose that the software behaves in a way that you don't consider appropriate. Does it matter whether the behavior conflicts with the specification? Why? **And** why not? (explain both)

S.4. Distinguish between customer satisfiers and dissatisfiers. Think of OpenOffice Impress: Give two examples of each.

S.5. Describe the case of *Family Drug Store v. Gulf States Computer*. (To prepare for this question, read the original case, not just the 1 slide summary.) Was the product defective? In what way(s)? Assume that it was defective--why was the lawsuit unsuccessful? What differences in the sales meeting might have made the lawsuit come out differently?

S.6. What are some advantages and problems with using "conforms to specifications" as an indicator of quality?

Long Answer

L.1. Compare and contrast Weinberg's definition of quality (quality is value to some person) and Crosby's (conformance to requirements). (Note: pay attention to the accuracy of your characterization of Crosby's work.)

L.2. Compare and contrast three definitions of "software quality." Which do you prefer? Why?

S.1. Compare and contrast three different definitions of "software error." Which do you prefer? Why?

Bug Advocacy

Objections to Bug Reports

Editing Bugs

Definitions

- coding error
- configuration-dependent failure
- corner case
- customer impact (of a bug)
- delayed effect bug
- defect
- enhancement
- error
- failure
- failure conditions
- fault
- interrupt
- memory leak
- priority (of a bug)
- race condition
- severity (of a bug)
- symptom
- wild pointer

Short Answer

S.1. Explain the analogy between sales and bug reporting. What is your opinion of this analogy? Why?

S.2. What is the difference between a fault and a failure? Given an example of each.

S.3. One reason often given for fully scripting test cases is that the tester who follows a script will know what she was doing when the program failed, and so she will be able to reproduce the bug. What do you think of this assertion? Why?

S.4. What is the difference between severity and priority of a bug? Why would a bug tracking system use both?

S.5. Why test on one computer and replicate tests on a different one? What would you do if the results were different?

S.6. Joseph Juran distinguished between Customer Satisfiers and Dissatisfiers as key aspects of quality. A common extension adds the idea of "Delighters" (satisfiers that go beyond customer expectations). We could extend this further with the concept of "Stinkers" (dissatisfiers that go beyond what customers think of as forgivable faults). Consider the product you are testing in this course: give an example, from your product, of each of these four things (satisfiers, dissatisfiers, delighters and stinkers) and explain why each is a good example.

S.7 Consider a bug report that a program prints a blank page at the end of a document when the document's last page is full. Describe two circumstances under which a programmer could reasonably argue that this bug

should not be fixed and explain what makes the arguments reasonable.

S.8 The lecture presented a 6-factor approach to bug reporting: RIMGEA. What are the factors? (List and briefly describe them. Use a simple example when appropriate.)

S.9. What does the lecture mean by a "credible source"? Suppose that you think it is really difficult to create appealing templates with OOo Impress. Who could be your credible sources for supporting (or reeducating you on) your design critiques?

S.10 Explain the idea that we make bug-related decisions under uncertainty. What do you think of it

Long Answer

L.1. Suppose that you find a reproducible failure that doesn't look very serious.

- Describe the four tactics presented in the lecture for testing whether the defect is more serious than it first appeared.
- As a particular example, suppose that the display got a little corrupted (stray dots on the screen, an unexpected font change, that kind of stuff) in Impress when you drag the scroll bar up and down. Describe four follow-up tests that you would run, one for each of the tactics that you listed above.

L.2. Suppose that you found a reproducible failure, reported it, and the bug was deferred. *Other than further testing*, what types of evidence could you use to support an argument that this bug should be fixed, and where would you look for each of those types of evidence?

L.2a. Suppose that you found a reproducible failure in the handling of tables by OOo Impress. You reported it, and the bug was deferred. *Other than further testing*, what types of evidence could you use to support an argument that this bug should be fixed, and where would you look for each of those types of evidence?

L.3. Imagine that you report something wrong with the program and the project manager responds that you have written an enhancement request, not a bug report. Assume that the project manager is making a rational statement that she believes is correct. How would she justify her statement (a) in a company that creates custom software under contracts that include detailed specifications or (b) in a company that develops software for its own use and does not rely on trusted specifications? In both cases, describe her strongest arguments. Then evaluate the arguments.

L.4. Describe a bug that you reported about a product that you are testing. Provide a cost-of-quality argument to justify fixing it, and a cost-of-quality argument to justify not fixing it. In both cases, provide the best argument you can think of. In the specific case of this bug, which argument is more persuasive, and why?

L.5. Describe the evidence in favor of the assertions that (a) late bug fixes are expensive and risky and (b) late changes to the external design of the program are expensive and risky. How could you revise the development process (including the testing process) to reduce the cost and risk of late changes? What new costs and risks would this process revision carry? Describe a context in which the revision would be worthwhile and explain why you think so. Describe a context in which it would be inappropriate and explain why you think so.

L.6. The Bug Advocacy lectures advise using test tools to improve bug reproducibility. Try one of the test tools that you think should help with reproducibility (obviously, you have to have done this before coming to the exam, but that's why we post the study guide in advance). Describe your experience with the tool and your sense of its strengths and weaknesses in terms of helping you improve the reproducibility of failing tests.

L.7. Joseph Juran distinguished between Customer Satisfiers and Dissatisfiers as key aspects of quality. In contrast, Gerald Weinberg defines quality as “value to some person.” Suppose we revised a program so that it looked up all display text (e.g. menu names) from a text file on disk, so the user could change the language of the program at any time (e.g. French menu text to Chinese). Suppose too that, because of this change, the new version is much slower than the last, making some people unhappy with it. How would Juran evaluate this (is it a bug, or a feature?) How would Weinberg? Explain your reasoning in both cases.

L.8. A bug has been in a product for 5 years. The project manager refuses to fix it, claiming that it has no importance because it has had no customer impact. Describe three ways that you could check whether this bug (or one like it) has customer impact. In each case, include in your description what kind of information you are looking for (be specific), how you would get that type of information, and how that information would be useful in demonstrating the magnitude of customer impact.

L.9. A program's memory management is weak. You have reported several crashes caused by out-of-memory conditions but the project manager responds by saying customers should buy more memory if they're going to do tasks that are so complex. To this point, the triage team agrees with the project manager. Suppose that you wanted to get this bug fixed. What are your three best ideas for influencing people in order to fix the bug? Explain why these are good ideas.

L. 10 Consider a bug report that a program prints a blank page at the end of a document when the document's last page is full. Describe three ways that a programmer could reasonably argue that this bug should not be fixed and explain what makes each argument reasonable. (Note: think back to your studies of oracle heuristics.)

L.11. The lectures draw an analogy between bug analysis and signal detection theory. What is signal detection theory and how has it been applied to complex judgments? (NOTE: Don't be one of the people who mistakenly asserts that SDT has only been applied in simple experiments like studies of sound perception. If you need more information, ask Google.) What guidance does SDT offer when we apply its reasoning to testing?

L.12 Describe three things that will bias a bug report reader against taking the bug report seriously. Why would these have that effect? How should you change your bug reporting behavior to avoid each of these problems?

L. 13 Consider three of the common sources of bias in decision-making: desired result, probability of the result, and perceived importance of the result. Now imagine managing a tester who you want to test the creation of templates for OpenOffice presentations. How might consideration of these biases guide your thinking about how to manage and advise this tester? (One often-successful way of approaching this question is to consider two different possible desired bug-reporting approaches (or patterns of bug-reporting behavior) for this tester, and for each, consider how setting your tester's bias in a certain way might foster one approach or the other.)

Quality Costs

Definitions

- Appraisal costs
- Cost of quality
- External failure costs
- Externalized failure cost
- Internal failure costs
- Prevention costs
- Quality / cost analysis

Short Answer

S.1. Why are late changes to a product often more expensive than early changes?

S.2. Compare, contrast, and give some examples of internal failure costs and external failure costs. What is the most important difference between these two types of failure cost?

S.3. What are some advantages and problems with using "conforms to specifications" as an indicator of quality?

Long Answer

L.1. Why are late changes to a product more expensive than early changes? How could we make them (late changes) cheaper?

L.2. Describe a bug that you reported about a product that you are testing. Provide a cost-of-quality argument to justify fixing it, and a cost-of-quality argument to justify not fixing it. In both cases, provide the best argument you can think of. In the specific case of this bug, which argument is more persuasive, and why?

L.3. Describe the evidence in favor of the assertions that (a) late bug fixes are expensive and risky and (b) late changes to the external design of the program are expensive and risky. How could you revise the development process (including the testing process) to reduce the cost and risk of late changes? What new costs and risks would this process revision carry? Describe a context in which the revision would be worthwhile and explain why you think so. Describe a context in which it would be inappropriate and explain why you think so.

L.4. What is the strongest ethical argument in favor of doing a cost-of-quality analysis that evaluates only cost to the vendor and does not estimate cost to the customer? What is the strongest ethical criticism? Evaluate both.

Credibility & Mission of the Bug Tracking Process

Definitions

- False alarm

Short Answer

S.1. Explain the idea that we make bug-related decisions under uncertainty. What do you think of it?

Long Answer

L.1. The lectures draw an analogy between bug analysis and signal detection theory. Explain the analogy. What are some strengths and weaknesses of this analogy?

L.2. Describe three things that will bias a bug report reader against taking the bug report seriously. Why would these have that effect? How should you change your bug reporting behavior to avoid each of these problems?

L.3. Consider three of the common sources of bias in decision-making: desired result, probability of the result, and perceived importance of the result. Now imagine managing a tester who you want to test the creation of templates for OpenOffice presentations. How might consideration of these biases guide your thinking about how to manage and advise this tester? (One often-successful way of approaching this question is to consider two different possible desired bug-reporting approaches (or patterns of bug-reporting behavior) for this tester, and for each, consider how setting your tester's bias in a certain way might foster one approach or the other.)

Measurement Theory

Code Coverage

Measuring the Extent of Testing

Definitions

- Attribute to be measured
- Balanced scorecard
- Code coverage (as a measure)
- Construct validity
- Defect arrival rate
- Defect arrival rate curve (Weibull distribution)
- Extent of testing
- Line (or statement) coverage
- Measurement
- Measurement error
- Measuring instrument
- Path coverage
- Side effect of a measurement
- Scope of a measurement
- Statement coverage
- Surrogate measure
- Validity of a measurement
- Weibull reliability model

Short Answer

S.1. Give two examples of defects you are likely to discover and five examples of defects that you are unlikely to discover if you focus your testing on line-and-branch coverage.

S.2. Distinguish between using code coverage to highlight what has not been tested from using code coverage to measure what has been tested. Use an example to make your contrast clearer.

S.3. What is the Defect Arrival Rate? Some authors model the defect arrival rate using a Weibull probability distribution. Describe this curve and briefly explain three of the claimed strengths and three of the claimed weaknesses or risks of using this curve.

S.4. What is a model? Why are models important in measurement?

S.5. How is it that you can achieve very high coverage from your tests but still miss lots of bugs?

S.6. Why is it usually impossible to achieve complete path coverage? Use examples to clarify your answer.

Long Answer

L.1. Explain two advantages and three disadvantages of using bug counts to evaluate testers' work.

L.2. Consider the following defect removal bar chart. Write a brief critique of the project. Do you think the software was delivered late? Why or why not? The chief software engineer was assigned to a new project after one of the reviews. Which one?



L.3. Distinguish between using code coverage to highlight what has not been tested from using code coverage to measure what has been tested. Describe some benefits and some risks of each type of use. (In total, across the two uses, describe three benefits and three risks.)

L.4. A company pays bonuses to programmers for correcting bugs. The more bugs you fix, the bigger the bonus. This includes bugs coming from the customer and from the system test group, and it includes bugs actually made by the programmer who fixes them.

The underlying measurement in this case is the Bug Correction Count (BCC), the number of bugs fixed during this pay period.

Suppose that we published Bug Correction Counts (BCC), for each programmer, for each pay period, over a period of one year. Any manager in the company could review this data.

- What's the measurement scale of BCC?
- What attribute (if any) does BCC measure?
- List three attributes that you think someone might use BCC to attempt to measure.
- For each way that you think BCC might be used, apply the 10 point validity analysis.

L.5. Given that complete testing is impossible, what are some potential stopping rules (rules for deciding when to stop testing)? Describe the rules you propose, and for each, explain the tradeoffs involved in using it (what's good about it, what's bad about it, what risks does it introduce, etc.).

Status Reporting

Definitions

- Milestones
- Status report

Short Answer

S.1. Describe three indicators of project status.

Long Answer

L.1. Describe four indicators of project status and give examples of each.

Career planning

Definitions

- Black box tester
- Consultant
- Contractor
- Software process improvement specialist
- Subject matter expert
- Systems analyst
- Test automation programmer
- Test automation architect
- Test lead
- Test manager
- Test planner
- User interface critic

Short Answer

- S.1. Distinguish between technical and management types of roles in a software testing group. Give two examples of each.
- S.2. Distinguish between technical and process management types of roles in a software testing group. Give two examples of each.
- S.3. What is the Best Alternative to a Negotiated Agreement? How do you use one of these in the course of a job search?

Long Answer

Recruiting New Testers

Definitions

- Behavioral interviewing
- Essential job functions
- Job description
- KSAO
- Opportunity hire

Short Answer

- S.1. What is an opportunity hire? What benefits and risks are associated with opportunity-hiring?
- S.2. The notes state that "It is a more serious mistake to hire badly than to pass up a good candidate." Contrast the problems associated with a bad hire to those associated with passing up a good candidate. Which set of problems are worse? Why?
- S.3. What is the primary goal of the job interview?
- S.4. What are some of the most effective ways of structuring a job interview?

Long Answer

- L.1. The notes say that "Diversity is essential." Why is this? Why not fill the group with experienced testers who all have programming skills? Give examples to support your points and arguments.

Learning Styles and Testing

Definitions

- Bloom's taxonomy
- Analysis (in Bloom's taxonomy)
- Application (in Bloom's taxonomy)
- Comprehension (in Bloom's taxonomy)
- Critical thinking skills
- Evaluation (in Bloom's taxonomy)
- Knowledge (in Bloom's taxonomy)

- Synthesis (in Bloom's taxonomy)

Short Answer

Long Answer

Outsourcing

Definitions

Short Answer

Long Answer

Legal Issues

Definitions

- Negligence

Short Answer

Long Answer

L.1. Vendor sells custom software with a development contract that promises the applications will be "completely tested." In fact, the testing done by Vendor includes complete statement and branch coverage. Vendor delivers a product to Customer, the product corrupts its data without indicating any problem, and Customer loses \$2 million. Customer sues, arguing that no product that has a serious bug (you should assume in your answer that this is, in fact, a serious bug) could have been completely tested.

Vendor responds with two arguments: First, that complete statement and branch coverage *is* complete testing. Second, that no reasonable customer could believe that a software contract would promise such extensive testing that they could guarantee bug-free software.

Your tasks:

- Describe Vendor's arguments in the most convincing (but accurate and honest) way that you can
- Evaluate the arguments.

Management Issues

Roles of Test Groups

Definitions

Short Answer

S.1. List (and briefly describe) three different missions for a test group. How would your testing strategy differ across the three missions?

Long Answer

L.1. Suppose that a test group's mission is to achieve its primary information objective. Consider (and list) three different objectives. For each one, how would you focus your testing? How would your testing differ from objective to objective?

Measuring Tester Performance

Definitions

Short Answer

Long Answer

Five (Plus) Schools of Software Testing

Definitions

Short Answer

Long Answer

Context Analysis

Definitions

Short Answer

Long Answer

Research Guide to Testing

Definitions

Short Answer

Long Answer

Introduction to Test Driven Development

Definitions

- change detector
- refactoring
- unit testing

Short Answer

S.1. Contrast the maintenance costs of unit-level regression suites and GUI-level regression.

S.2. Contrast the reporting and troubleshooting costs of unit-level regression testing and GUI-level regression.

Long Answer

Lifecycle Models and their Implications for Testing

Definitions

- Evolutionary development
- Lifecycle model
- Waterfall lifecycle

Short Answer

S.1. List and briefly explain three strengths of the waterfall lifecycle.

S.2. List and briefly explain three strengths of the evolutionary lifecycle.

Long Answer

L.1. Compare the evolutionary and waterfall lifecycle models. Consider the four factors that project managers have to trade off against each other, and any additional issues (1 to 3 of them) that you think are important.

L.2. Consider the spiral lifecycle. Describe the tradeoffs in following this, compared to evolution or waterfall in terms of the four factors that project managers have to trade off against each other.

Testability

Definitions

Short Answer

Long Answer

Testers' Notebooks

Definitions

Short Answer

Long Answer

Project Resource Estimation

Definitions

Short Answer

Long Answer

Candidates for new questions

TROUBLESHOOTING (from a discussion forum): We are using Rational ClearQuest to log defects. If the system is not available to run any transaction, is that a defect or an incident. If I open the application and it gives an error which means the system is not available. Should that be logged in Defect tracking tool or should that be treated as an incident and followed up outside the Clear Quest?

Note the cool issues:

- Is the unready system the bug reporting system or the environment under test? (Why is it relevant that they are using clearquest in this question?)
- Is the server a production server or one dedicated to testing the application under test?
- Does “any transaction” mean no transaction on the system as a whole or just in this application?
- Do other applications give a message that the system is unavailable?
- Is the system software part of what the company is developing or is it merely the environment under which the application is run?

Copyright (c) Cem Kaner 2002-2008

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
