

Grading Essay Exams in the Black Box Software Testing Course

Cem Kaner, J.D., Ph.D.

September, 2006

Copyright (c) Cem Kaner

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.

What's in this series

This series is a set of modules that describe my approach to grading essays by example.

1. Overview of my grading process
2. Example 1 (long essay)
3. Example 2 (long essay)
4. Example 3 (short essay)

Grading Essay Exams in the Black Box Software Testing Course

Part 1: Overview of my Grading Process

Cem Kaner, J.D., Ph.D.

September, 2006

Copyright (c) Cem Kaner

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.

Why do instructional assessment?

Instructional assessment provides four important benefits:

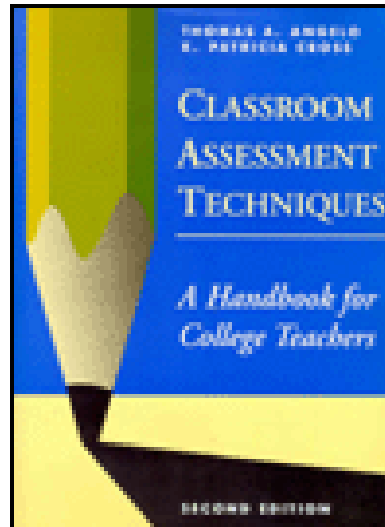
- Feedback to the student about what s/he knows or does not know
- Incentive to study certain things, to study them in certain ways, to master certain skills
- Feedback to the instructor about what the students are learning and not learning
- Data to be used in determining a grade or making a pass/fail decision.

The pass-fail decision might be the most important or the least important aspect of your grading.

Industrial classes, for example, cry out for feedback, but not for recorded grades.

A few useful references

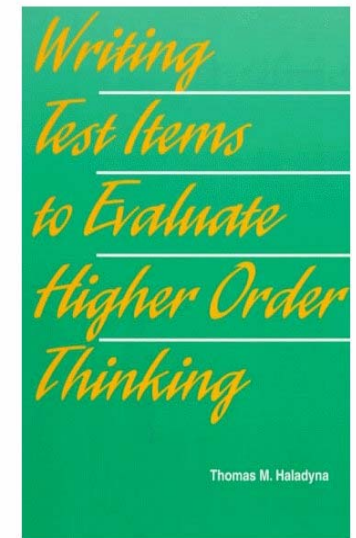
Angelo & Cross:
Classroom Assessment
Techniques



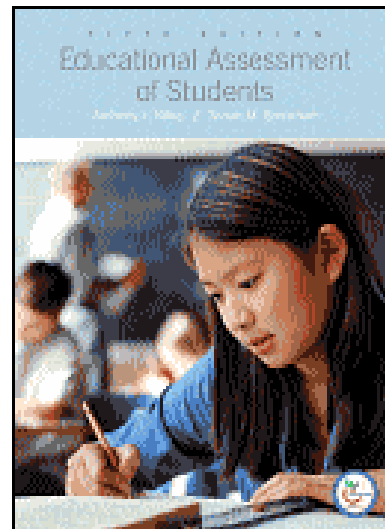
Arter & McTighe: Scoring
Rubrics in the Classroom



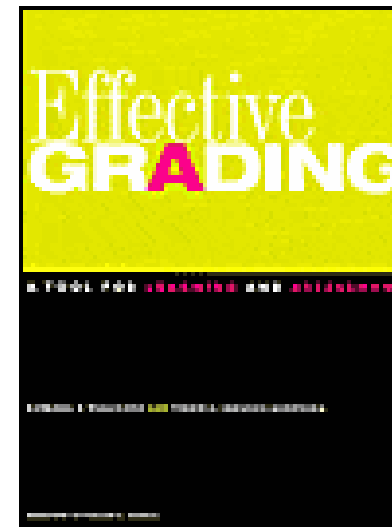
Haladnya: Writing Test
Items to Evaluate Higher
Order Thinking



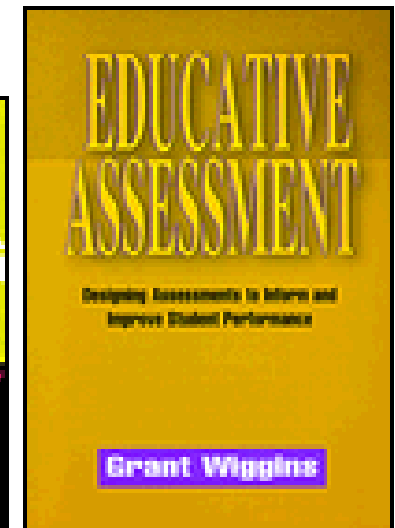
Nitko: Educational
Assessment of Students



Walvoord & Anderson:
Effective Grading



Wiggins: Educative
Assessment



My examination process

- I create the study guide from a large pool of essay questions
- The study guide has 100 questions (50 short essay, 50 long essay)
- Students get this in the first or second week of class
- All questions on all exams are drawn from this study guide

My expectations of the students

1. I expect students to prepare answers in advance.
2. I expect students to compare notes, reviewing each other's work.
 - I encourage this by hosting study sessions at school and at a local cafe
3. I expect students to be able to recognize a question, quickly remember their answer to it, and start writing.
 - Therefore, I allow little think time on the exam.
 - My rule of thumb is to allow them twice as long as it takes me to write out the answers.
4. I expect answers that are well thought out and well organized. ***I want to give good grades for good work.***

My expectations of the students

5. I expect good work from students whose first language is not English.
 - The students have plenty of time before the test to read the questions, translate them, talk them over with friends, get answers reviewed by other students and so on.
 - They have no excuse for not understanding the question
 - They have no excuse for writing unintelligible answers because they have time to get peer review.

I lay out ground rules in the study guide (1)

These are from my study guide. Yours might be different:

1. If a question contains multiple parts, you must provide a distinct answer for each part.
2. If a question asks about “some,” that means at least two. I normally expect three. Similarly if the question asks for a list, I expect a list of at least three.
3. Be aware that different words in questions have different meanings (see www.indiana.edu/~wts/wts/essayexam.html). For example, each of the following words calls for a different answer: identify, list, define, describe, explain, compare, contrast. If I ask you to list and describe some things, give me a brief identification (such as a name) of each (that’s the list) **and then** a description for each.

I lay out ground rules in the study guide (2)

4. If you find it hard to define or describe something, try writing your answer around an example.
5. To describe the relationship among things, you might find it easiest to work from a chart or a picture. You are not **required** to use a diagram unless I ask for one.
6. If I ask you to describe or define something that is primarily visual (such as a table or a graph), your answer will probably be easier to write and understand if you draw an example.
7. If I ask you for the result of a calculation (such as the number of paths through a loop) show your calculations or explain them. **Help me understand how you arrived at the answer.**

I lay out ground rules in the study guide (3)

8. If I ask you to analyze something according to the method described in a particular paper or by a particular person, I expect you to do it their way. If I ask you to describe their way, do so. If I ask you to apply their way, you don't have to describe it in detail, but you have to do the things they would do in the order they would do them, and you have to use their vocabulary to describe what you are doing.
9. **Don't answer what has not been asked.** For example, if I ask you to define one thing, don't define that and then give me the definition of something related to it. If you do,
 - (a) I won't give you extra credit,
 - (b) I'll think that you don't know the difference between the two things, and
 - (c) if you make a mistake, I'll take off points.

The study guide is a contract

- I refer to the study guide in class
- I show students how the study guide applies to their exam questions
 - This video is an example of that. I am designing it for instructors but share it with students.
- You can and should enforce the grading guidelines
 - If you don't mean them, don't say them
 - If you aren't willing to be bound by them, don't say them

My approach to grading

1. I review the question to identify its components
2. I tentatively outline answers for each component
 - I allow for the possibility that conflicting answers will be acceptable. Even an answer that I disagree with may demonstrate clear learning of directly relevant material.
3. I then assign grading weights to the components

My total across grading weights often exceeds 90% to allow for excellence in one component counterbalancing adequacy in another. (However, I typically cap the computed total at 90%, leaving 10% for clarity/organization.)

4. I reserve 10% of the grade (in U.S. grading, this is the difference between an A and a B) for clarity, organization, and my subjective assessment of the need for an adjustment to more accurately reflect my sense of the quality of the answer.

My approach to grading

5. If this is the first time I am grading or if the course materials have changed, then my expectations probably won't match what I will get from the students.
 - Based on what I see in the answers, I will often revise my outline and my weights.
 - This is NOT to make the grading easier or to make the average score higher. It is to give a fairer result.
6. I never grade on a scale. If everyone writes a poor answer, they all get a poor result. Accurate feedback is very important. Adjustment of final grades in the course is something that can be handled separately if needed.

Grading Essay Exams in the Black Box Software Testing Course

Part 2: Grading a long essay question (test automation)

Cem Kaner, J.D., Ph.D.

September, 2006

Copyright (c) Cem Kaner

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.

Example 1 - Long answer

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.

My thanks to Scott Barber for synthesizing the exemplars that follow from many actual student answers.

Original answers often had spelling or grammar mistakes that I ignored in grading and that are fixed here so that they don't distract you when you read them.

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.

STEP I. Identify the components

- There are two components to this question:
 - 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.

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.

STEP 2. Tentatively outline:

- Part 1 -- Why is it important to design maintainability into automated regression tests?
- Part 2 -- Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.

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.

STEP 2 (continued). Tentatively outline:

PART I: *Why is it important to design maintainability into automated regression tests?*

I see two themes:

- Change is natural, and therefore maintenance is necessary
- Maintainability is needed to reduce the cost of coping with change

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.

STEP 2 (continued). Tentatively outline:

PART I: Why is it important to design maintainability into automated regression tests?

I see two themes:

- Change is natural, and therefore maintenance is necessary
- Maintainability is needed to reduce the cost of coping with change

To help me think about grading this question I made a list of several ideas that illustrate each theme.

- I generated some myself
- I added good student ideas from exams
- My current way of grading (in themes) grew out of my analysis of the list.

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.

STEP 2 (continued). Tentatively outline:

PART 2: Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.

- I expect to see suggestions that tie to design of test code.
 - Students have already learned magic phrases for maintainability, like “be modular” and “use source control” that help in maintainability
 - I expect to see **design choices** (source control is not a design choice) that are specifically applied to test automation

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.

STEP 2 (continued). Tentatively outline:

PART 2: Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.

- A well written design suggestion (Part 2) often explicitly links to a maintainability issue the student raised in Part I.

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.

STEP 3. Assign grading weights

I weighted each part equally, and allowed up to 12 points each.

- I max the total at 90% (18/20), preserving 2 points (clarity).
- This creates flexibility. A strong answer to one part can be less impressive on the other and still get full points.
- This respects reasonable individual differences in how people interpret what a question like this is asking for.

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.

STEP 3. Assign grading weights

For the first part, I worked from the themes, assigning up to 6 points for coverage of each theme.

- This is not what I did initially. I started by allowing up to 4 points per reason for up to 3 reasons.
 - It was tedious to grade
 - There were often more than 3 reasons, and they were often related or redundant so it was hard (and inaccurate) to score them separately.
 - I adopted the overall score per theme after regrading exams from several years
- Students who made good points beyond these themes were rare and usually got extra points

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.

STEP 3. Assign grading weights

For the second part (design choices that foster maintainability):

- I looked for three ideas, allocating up to 4 points each
- I gave up to 2 points for a generic idea that was not directly tied to maintainability of test automation
- If the student offered more than 3 ideas, I picked the best 3 but downgraded if there were errors in the others.
 - A typical downgrade is -1 point.

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.

STEP 4. Assign discretionary points

I reserve 10% of the grade for my discretionary, subjective assessment of the answer.

- **Clarity** is a factor. If I find it hard to understand an answer (but finally figure it out), this is where I impose a penalty
- **Structure** is a factor. Many students merge the answer to all parts into one mess. They get zero for structure.
- **Clue** is a factor. If I develop the subjective impression that this student has more of a clue about this answer than my grading scale reflects, I can adjust up by 10%.

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1					
Student 2					
Student 3					

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.

So, let's
do some
grading!

Importance of designing maintainability into automated regression tests:

- True ROI from automated regression tests is achieved in versions following the one in which tests were automated. Therefore, maintainability is essential.
- Changes are made to the product in every build. Unmaintainable tests will either break, become obsolete or become too expensive to update.
- If the automated regression tests are not maintainable, mismatches between test docs and test code will lead to further ambiguity and bugs.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code.
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained in multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

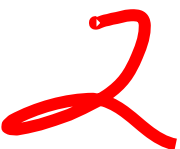



Let's focus on each part in turn ...

Importance of designing maintainability into automated regression tests:

- True ROI from automated regression tests is achieved in versions following the one in which tests were automated. Therefore, maintainability is essential.
- Changes are made to the product in every build. Unmaintainable tests will either break, become obsolete or become too expensive to update.
- If the automated regression tests are not maintainable, mismatches between test docs and test code will lead to further ambiguity and bugs.

Let's focus on each part in turn ...

Importance of designing maintainability into automated regression tests:

- True ROI from automated regression tests is achieved in versions following the one in which tests were automated.  Therefore, maintainability is essential.
- Changes are made to the product in every build.  Unmaintainable tests will either break, become obsolete or  become too expensive to update.
- If the automated regression tests are not maintainable, mismatches between test docs and test code will lead to further ambiguity and bugs. 

Theme 1: change is inevitable

Theme 2: cope with cost of change

a different issue from most students. Good point

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.

Work from the grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12				
Student 2					
Student 3					

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code.
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained in multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK – a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained in multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK – a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained ~~in~~ multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK - a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained ~~in~~ multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets. *Ties the first 2 to testing*
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK – a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained ~~in~~ multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets. *But this is wrong*
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain.

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK – a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained ~~in~~ multiple places.
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain. *GOOD*

Some design choices will make automated regression tests more maintainable:

- Parameterize the code – rather than hard-coding values that may change over time, parameterize variables and store their values in a file external to the code. *OK – a bit generic*
- Modularize the code – put commonly executed steps or processes in callable subroutines. This protects against redundant code that must be maintained in multiple places. *OK*
- Use a data-driven approach – Don't embed all the data. Store the data, actions to perform on the data, and actions in the code/scripts of the regression tests in external objects (possibly a spreadsheet). Any change made to the program would only need to be updated in the spreadsheets.
- Do not rely on code generated by a capture-replay tool. These tools tend to generate fragile scripts that are extremely difficult to maintain. *GOOD*

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2					
Student 3					

What comments do I put on the page?

Comments are

- For the student, OR
- For you, in case the student questions you, OR
- For you, notes as you grade (and regrade), OR
- For explaining your grading to someone else (e.g. another teacher).

My notes today were for explaining to another teacher.

My not-for-student notes, I don't show the student.

My notes to students are generally brief.

- I might share the chart (their row on the chart)
- I review issues across the full class

Now, let's look at a weaker answer to the same question

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.

Second sample answer

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test. If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain.

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code.

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

Part 1: Why is it important?

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test. If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain.

Part 2: Design choices

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code.

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

Part 1: Why important

ROI Problem?

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. *OK*

For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test.

If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain.

Part 1: Why important

ROI Problem?

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. OK

For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test. Good

If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain.

Part 1: Why important

ROI problem?

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. OK

For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test. Good

If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain.

This clarifies part of the first point, but what is the ROI problem?

Part 1: Why is it important?

One of the keys to achieving a good ROI with an automated regression test is that it must be reusable. **2 Weak**

For the test to remain effective across multiple builds or versions of a program, it is critical that the test be designed to easily incorporate the inevitable changes that will occur in the system under test. **1 good**

If a test is not designed with maintainability in mind, the test will quickly become obsolete or too costly to maintain. **2 OK**

Theme 1: change is inevitable
Theme 2: cope with cost of change

5
4 ? 3 to 5

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9				
Student 3					

Part 2: Design choices (code design)

OK - generic

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code.

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

Part 2

OK - generic

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code. *Excellent*

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

Part 2

OK - generic

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code. *Excellent*

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

oops !!

Part 2

Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.

This is NOT a choice for design of the test code.

- Therefore it is irrelevant and gets (at best) no credit
- If there are errors, I take points off

Grading Part 2: up to 4 points for each design choice

One of the most important design choices for an automated test suite is modularity. This allows similar tests to be updated by changing the code in one place rather than many places.

3

Possibly the second most important design choice is variable parameterization and data abstraction. This allows future testers to edit the parameters and test data in a file that is separate from the code. 4

Regardless of the test design decisions, for an automated regression test to be maintainable, it must include well-organized documentation that helps future testers to quickly pinpoint where to look in the code when a certain change needs to be made.



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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9	7	16	0.5	16.5
Student 3					

Here's a third (weaker) answer to the same question

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.

Third sample answer

Maintainability is important for automated regression tests because if not easily maintained, minor changes in the system under test can cause the tests to become valueless or misleading. Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed.

Documentation – good documentation points directly to what needs to be changed, how and why.

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes.

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer

Maintainability is important for automated regression tests because if not easily maintained, minor changes in the system under test can cause the tests to become valueless or misleading. Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed.

Documentation – good documentation points directly to what needs to be changed, how and why.

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes.

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer – Part 1

Maintainability is important for automated regression tests because if not easily maintained, minor changes in the system under test can cause the tests to become valueless or misleading.

Third sample answer – Part 2

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed.

Documentation – good documentation points directly to what needs to be changed, how and why.

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes.

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer - Part 1

Maintainability is important for automated regression tests because if not easily maintained, minor changes in the system under test can cause the tests to become valueless or misleading.

Theme 1 → 0

Theme 2 → generic, vague,
obvious from
definition of
"maintainability"

Theme 1: change is inevitable

Theme 2: cope with cost of change

→ 4

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9	7	16	0.5	16.5
Student 3	4				

Third sample answer - Part 2

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed. ? OK ?

Documentation – good documentation points directly to what needs to be changed, how and why.

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes.

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer - Part 2

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed. *? OK ?*

Documentation – good documentation points directly to what needs to be changed, how and why. *Code design ?*

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes.

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer - Part 2

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed. *? OK ?*

Documentation – good documentation points directly to what needs to be changed, how and why. *Code design ?*

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes. *generic*

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed.

Third sample answer - Part 2

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed. *? OK ?*

Documentation – good documentation points directly to what needs to be changed, how and why. *Code design ?*

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes. *generic*

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed. *CODE design*

Grading Part 2: up to 4 points for each design choice

Some important design choices to consider when creating an automated regression test suite include:

Modularity – the more abstract and cohesive the parts of the test, the fewer things will have to be changed. ~~2~~ ???

Documentation – good documentation points directly to what needs to be changed, how and why. ~~0~~

Information hiding – the less the code “knows” about the other code it uses, the less affected it will be if the other code changes. ~~2~~

Traceability - to the spec and to the code, the easier it is to see what is being tested and why, the easier it is to understand what needs to be changed. ~~0~~

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9	7	16	0.5	16.5
Student 3	4	4	8	1	9

Here's a fourth (shotgun) answer

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.

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. Maintaining an automated regression test is very important because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

Design choices:

1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool
3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 1

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. Maintaining an automated regression test is very important because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

Fourth sample answer – Part 2

Design choices:

1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool
3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

What's on topic?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. Maintaining an automated regression test is very important because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. — *OK, setting up the topic*

Automating the regression testing involves a huge amount of work. Maintaining an automated regression test is very important because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. Maintaining an automated regression test is very important because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

→ automation is expensive

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. ~~Maintaining an automated regression test is very important~~ because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

→ automation is expensive

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. ~~Maintaining an automated regression test is very important~~ because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

→ automation is expensive

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. ~~Maintaining an automated regression test is very important~~ because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.

→ automation is expensive

Fourth sample answer - Part 1

Why is it important to design maintainability into automated regression tests?

Maintenance changes for automated regression tests are made after a change is made to the code. Automating the regression testing involves a huge amount of work. ~~Maintaining an automated regression test is very important~~ because a small change by a programmer can cause serious problems while performing GUI tests. Building an automated tool for testing is a complex process. It involves time and huge design constraints and thinking. Once built, using automated regression tests, a lot goes into using it. Also, it should be very generic so that one should not struggle in giving test cases for different applications. ~~Therefore it is wise to have maintainability for automated test tools, so that one can make changes and make it usable for different regression tests on different products and applications.~~

→ automation is expensive

Grading part 1

Theme 1 – change is inevitable – *none*

0

Theme 2 – cope with cost of change

- automation is expensive 4
- small programmer changes can "cause serious problems" ?

His or her added theme – extension to new products

- create opportunity to use the tool (?) (tests ?) in other tests of other applications 3

Tie this together into a grading chart

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.

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9	7	16	0.5	16.5
Student 3	4	4	8	1	9
Student 4	7 generous				

Fourth sample answer - Part 2

*Pure
shots on*

Design choices:

1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool
3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 2

Pure
shots on

Design choices:

- No 1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool
3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 2

Pure
shotgun

Design choices:

- No 1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool ? Huh? ... in shotgun ...
3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 2

Pure
shotgun

Design choices:

- No 1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool ? Huh? ... in shotgun ...
- No 3. Regression tests should be designed so there is no ambiguity in the results of the test.
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 2

Pure
shotgun

Design choices:

- No 1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool ? Huh? ... in shotgun ...
- No 3. Regression tests should be designed so there is no ambiguity in the results of the test.
- How? 4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

Fourth sample answer - Part 2

Pure
shotgun

Design choices:

- No 1. A proper user interface has to be considered and developed
2. A more generic approach should be considered in designing an automated regression tool ? Huh? ... in shotgun ...
- No 3. Regression tests should be designed so there is no ambiguity in the results of the test.
- How? 4. A good design would further enhance the opportunity for the tool to be adapted to different test applications
- No 5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored.

"No" = "no relevance to design of code for maintainability."

Grading the Fourth sample answer - Part 2

Design choices:

1. A proper user interface has to be considered and developed ~~0~~
2. A more generic approach should be considered in designing an automated regression tool. 0.5
3. Regression tests should be designed so there is no ambiguity in the results of the test. 0
4. A good design would further enhance the opportunity for the tool to be adapted to different test applications 0.5
5. Also, because regression testing involves bug regression, we should ensure that previous test results are stored. 0

Total = 1/12

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.

Tie this together into a grading chart

Student name	Why important	Describe design	Sub-total	Discretion	Total score
	2 @ 6 = 12	3 @ 4 = 12	18	2	20
Student 1	12	10	18	2	20
Student 2	9	7	16	0.5	16.5
Student 3	4	4	8	1	9
Student 4	7 generous	1	8	0	8

Grading Essay Exams in the Black Box Software Testing Course

Part 3: Grading a long essay question (test automation)

Cem Kaner, J.D., Ph.D.

September, 2006

Copyright (c) Cem Kaner

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.