My Report

Last Modified: 10/21/2015
Completion Status: Completed

1. D1. How many years have you been programming?

Text Response
7
7
7
8
10
6
7
5
7
10
5
5
8
7
6
5
15
6
5

Statistic	Value
Total Responses	19

2. D2. What is your highest degree level achieved?

#	Answer	Bar	Response	%
1	High school		0	0%
2	Bachelor		8	42%
3	Master		10	53%
4	Ph.D.		1	5%
	Total		19	

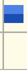
Statistic	Value
Min Value	2
Max Value	4
Mean	2.63
Variance	0.36
Standard Deviation	0.60
Total Responses	19

3. D3. How many years of industrial/open source experience do you have (programmer, software engineer, etc.)?

Text Response
0
6
0
0
2
0
4
0.5
3
2
1
0
0
0
0
0
0
4
1

Statistic	Value
Total Responses	19

4. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		13	68%
2	The description misses some important information to understand the unit test case.		5	26%
3	The description misses the majority of the important information to understand the unit test case.		1	5%
	Total		19	

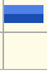
Statistic	Value
Min Value	1
Max Value	3
Mean	1.37
Variance	0.36
Standard Deviation	0.60
Total Responses	19

5. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		11	58%
2	The description contains some redundant/useless information.		7	37%
3	The description contains a lot of redundant/useless information.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.47
Variance	0.37
Standard Deviation	0.61
Total Responses	19

6. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		12	63%
2	The description is somewhat readable and understandable.		5	26%
3	The description is hard to read and understand.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.47
Variance	0.49
Standard Deviation	0.70
Total Responses	19

7. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
Too much information
Again, the description covers all the relevant information. The second validation is a little redundant as the variable depends on many of the same variables from the first validation.
For Q3, There is one sentence "this unit test case includes following focal methods" however it's only one focal method that is involved. So maybe we could show users "This unit test case includes #number# focal method#s#".
The way result is obtained is given twice.
It provides all the information needed and then some. This, of course, hurts accuracy and expressiveness.
The source here is again missing line numbers, although I realize now I may have misunderstood the question. To that end, I might actually say that my response to this question on the previous test case ought to have been that the description did not miss any important information. Here, though, I feel like some important information is missing. The unit test case must validate the value of \`statusCodeResult.statusCode\`, although the description itself does not reveal where this originates, as is teh case with \`result\`, which is tested in another assertion.
The information about the slicing path doesn't provide information about how exactly the other variable affect main variable.
Instead of saying result is obtained from 1) variable resource through slicing path It would be more helpful to replace "slicing path" with the actual code or have the popover when hovering over "slicing path" with the actual line of code.
Q1: for focal methods, may add var statusCodeResult = result as StatusCodeResourceResult; or just remove all " This unit test case includes following focal methods" description. Q2 and Q3 : no need explanation, description is concise and easy to read and understand.
Same as the former one
While the description captures all the silent information from the Unit Test, the general description at the top could be improved, as it is now, it seems to parse the method name and encapsulate that in the description.
The description is consistent with the code. And it is easy to understand and to trace the source.
The message provides a genealogy for each of the assertions, and the genealogy is broken up into readable sections.
Again, the summary sentence (i.e. "This unit test case method tests...") ideally should be a complete, grammatically correct English sentence. It is however, clear and easy to understand as-is.
Again, "This unit test case method tests reject invalid request id parameters . " is not a meaningful sentence. "(1) var result = resource.Execute(contextMock.Object,configMock.Object);(@line 78)" identifies a line of code/statement, but not a method as indicated in the headline. Referencing line numbers in the text does not help much. It does not become clear how 'statusCodeResult' is connected to 'result'. Due to that, and because the code is not complex and shorter, again, it is much easier to read the code.
The description of the focal methods can be easily misinterpreted because it is not clear that the class that is being tested is PopupResource, and that the method under test is contained in such class. Yeah, the description is shown in a section labeled with a title that contains the work PopupResourceShould; however, I would say that the suffix "should" is not a real convention to describe a test class. Actually, I had to review several times the entire description to understand it. At the end, it was easier to read the code. The names of the variables do not offer enough information about its purpose. For example, resource.execute() provides the same information than myvar.execute() or foo.execute(), but if (PopupResource)resource.execute is shown, the purpose/location of the variable would be immediately clear. The same thing is happening with the output of the focal method. In this case, "var result" does not offer enough information. I understand that the language is dynamically typed so there is no need to mention the output class in the unit test. However, in order to comprehend the test, I would prefer something like "IResourceResult result = ", which can be easily extracted from the source code. I would be awesome if the description points out that the test case is using mocks and not the actual implementation of some of the objects involved in it. Yeah, there are a couple of variables with the suffix mock, but if the programmer decides to ignore the convention (e.g. he names a mocked object as "foo"), it would not be clear that the test in fact uses mocks. Again, all those slicing paths do not provide enough information by their own. I had to read the code every time to understand each path. Perhaps the problem lies in the fact that the slicing path is showing only the name of the variables and not their types. The description is nicely written.
The description contains a lot of detail about the unit test case. I don't believe all of this information is necessary to understand the main idea. The details are best understood only when looking at the code. The first line was helpful in determining on a high level what was done, the details were more low level than may be needed.

Statistic	Value
Total Responses	17

8. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		4	21%
3	The description misses the majority of the important information to understand the unit test case.		1	5%
	Total		19	

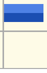
Statistic	Value
Min Value	1
Max Value	3
Mean	1.32
Variance	0.34
Standard Deviation	0.58
Total Responses	19

9. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		8	42%
2	The description contains some redundant/useless information.		10	53%
3	The description contains a lot of redundant/useless information.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.63
Variance	0.36
Standard Deviation	0.60
Total Responses	19

10. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		13	68%
2	The description is somewhat readable and understandable.		4	21%
3	The description is hard to read and understand.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.42
Variance	0.48
Standard Deviation	0.69
Total Responses	19

11. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
The description does not miss anything and is concise, but I would not say that is very readable. For an expert developer, or kind of (like me), it's even more clear and easier to just look at the actual code.
As every validation involves the same variable, a lot of redundant information is repeated due to describing how the variable is obtained.
For Q1 of this case I think it's miss some information. I believe the constructer needs three parameters: controllerContext, controllerContext and propertyDescriptor and we want to verify these three parameters are passed right. However, the result only says "This unit test case method tests construct controller context" which only contains one parameter. I think it is better to say "This unit test case method tests construct controller context, controllerContext and propertyDescriptor".
There too much information with not a lot of context.
All information required to understand the test cases is present. However, the variable `sut` is used in all three assertions; thus it is superfluous to state where it is obtained from in all three cases. This makes it mildly more difficlut to understand, but not enough to say it isn't easy.
In this case for all the three cases the slicing paths are the same.
Same as my previous responses. Once again the description next to the chat icon is unhelpful.
Q2: if all three test variables come from same place, may combine them into one paragraph. Q1 and Q3: no need explanation, complete and easy to ready and understand.
Q2: I think the three variables parts is a bit of redundant since all three vars in every section are all the same and come from a same source.
With these descriptions, I think the more useful aspect is the natural language description of what the test actually tests, as it give at a glance information about the test, which is great. However, the general overview description at the beginning could be improved, as in this case it didn't capture all of the information that the unit test deals with.
It is easy to understand the description. But in the validation part, the content under the 3 bullets are duplicate.
The description accurately describes the genealogy of the sut derived from three components. The information is redundant because the 3 variables are all linked but it verbosely repeats the same information for each.
I find myself wanting to visually see the slicing path appearing highlighting in the source code.
Same issues as before - code easier to read.
Honestly this is the first time that I see a unit test with parameters. Definitely I need to write and review more test cases, but I would say: 1) If the unit test is being called by other method (which is a weird thing for a *unit* test), I would need the information of such method to fully understand thie test case. The current description mentions the assertions between a sub.X and an arbitrary object X, but it is not clear how that arbitrary object X is created. 2) If the test case is not called by another method, I would like to see something suggesting that the arbitrary objects X's are received as parameters. The description mentions that there are 3 comparison between members of the sut and other objects. However, I am not sure if that information is enough to fully understand the test case. The slicing paths still need the types of the objects in order to provide enough information. The line numbers are really useful!! (I forgot to mention that) Finally, the description is nicely written.
The first line in the description misses model binding and property description of the construct. I think that the description is readable but the validations are a little redundant and tiring to read. Although it is very clear it is not as concise.

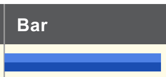
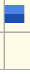
Statistic	Value
Total Responses	16

12. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		5	26%
3	The description misses the majority of the important information to understand the unit test case.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.26
Variance	0.20
Standard Deviation	0.45
Total Responses	19

13. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		8	42%
2	The description contains some redundant/useless information.		10	53%
3	The description contains a lot of redundant/useless information.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.63
Variance	0.36
Standard Deviation	0.60
Total Responses	19

14. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		12	63%
2	The description is somewhat readable and understandable.		5	26%
3	The description is hard to read and understand.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.47
Variance	0.49
Standard Deviation	0.70
Total Responses	19

15. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
The sliding path does help to understand where the result come from, but I don't really like how it is implemented right now. I would show the code and highlight the variables and the lines that form the slicing path!
Slicing path is not clear
The variables that aren't derived from other variables lack a description on how they are obtained. There is again redundant information when the same variable is in multiple assertions.
For Q2 the result shows paragraph "result is obtained from ..." twice. I think these two paragraph are exactly same and tester only needs to read it once.
How to obtain overrideMeAlternate is not described.
Again, same problem as previously noted. All needed information is there, but there is a lot of information that might not be needed at any given point in time. That makes it hard to read at a glance.
The description does not explain the origin of \`overrideMeAlternate\` and \`protectedOverrideMeAlternate\`. A quick glance at the code snippet shows that these are object created with a default constructor, and I think it is important for the description to show that in some way. This description suffers from the same problem as the previous in that the origin of the variable \`result\` is despalyed twice.
for result.HitCountOverrideMe and result.HitCountProtectedOverrideMe the slicing paths are the same
Same as previous responses.
Q2: if "this unit test case includes following focal methods" has already describe the var result. no more description for variables for result. Q1 and Q3: complete and expressive
Same as the former one.
I think adding some English context to the descriptions would be helpful to the readability. For instance, the description of this unit test could read "This unit test case method tests whether a call to ExtendClass creates a True Proxy.
same as before.
Similarly to the previous examples, it is concise and complete at the expense of some verbosity. I believe there may be an inherent tradeoff between redundancy and completeness. It's readable because it follows the same pattern
It describes the 4 assertions and provides a slicing path when necessary.
Same issues as before - code easier to read.
Again, the types of the variables would help me a lot to understand the test case. Also, those types can be added to the slicing paths. The message is nicely written.
It was a little difficult as I didn't completely understand what was happening in this section of code. The description captures well the public method name and it seems as though that is what's being tested but I'm unsure. The description doesn't make logical sense in the grammar of the sentence although it is fairly easy to get the gist of what is being said.

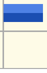
Statistic	Value
Total Responses	18

16. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		5	26%
3	The description misses the majority of the important information to understand the unit test case.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.26
Variance	0.20
Standard Deviation	0.45
Total Responses	19

17. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		8	42%
2	The description contains some redundant/useless information.		9	47%
3	The description contains a lot of redundant/useless information.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.68
Variance	0.45
Standard Deviation	0.67
Total Responses	19

18. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		14	74%
2	The description is somewhat readable and understandable.		3	16%
3	The description is hard to read and understand.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.37
Variance	0.47
Standard Deviation	0.68
Total Responses	19

19. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
the high-level idea of the test case is not clear
The variable "mixin" is derived from the proxy variable, which is not explicitly stated in the description. There is a lot of redundant information due to the same variables being used in multiple assertions and one being derived from the other.
This case is little more complicated. In this case we could see several previous problem happens together. The description only says "tests support mixins" however other variables except mixins also could affect our test result. In the "unit test case validates that: " part we also have some redundant information as last case.
The way to obtain proxy is redundant.
Overally verbose.
Again, the `proxy
Options 1 and 2 have the same slicing paths. Options 3 and 4 have the same slicing paths.
Previous comments hold. It's also becoming more apparent that there is a lot of redundant information. Where variable factory and dummyMixin come from is repeated for each assertion. This information only needs to be presented once.
Q1: in "This unit test case includes following focal methods:", may add " var mixin = proxy as IDummyMixin;". Q2: Also for the variable description, could delete those variable description. because they are related in focal method in Q1. Q3: easy to understand.
Q3: Even I could know the meaning of the description easily, I don't know if it is good enough NOT to treat mixin same as proxy (in the matter of focal method)
Same as before.
The descriptions captures the dependency for both proxy and Mixin on the the dummyMixin and that appears to be clear from the code
Easy to read, concise
Same issues as before - code easier to read.
From the description, it is not easy to understand what is the type of the variable "dummyMixing", which is a parameter of the focal method. I'm not familiar with this programming language (it looks like C#), but I would say that, in the code, there is a type cast from proxy to mixing. So the object in that case would be the same. However, the description suggests that proxy and mixing are different objects. Message is nicely written.
The description is clear on what is tested and captures well what is being done. The validation explanations may still be a little much but I suppose that is preference.

Statistic	Value
Total Responses	16

20. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		13	68%
2	The description misses some important information to understand the unit test case.		5	26%
3	The description misses the majority of the important information to understand the unit test case.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.37
Variance	0.36
Standard Deviation	0.60
Total Responses	19

21. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		6	32%
2	The description contains some redundant/useless information.		11	58%
3	The description contains a lot of redundant/useless information.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.79
Variance	0.40
Standard Deviation	0.63
Total Responses	19

22. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		9	47%
2	The description is somewhat readable and understandable.		7	37%
3	The description is hard to read and understand.		3	16%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.68
Variance	0.56
Standard Deviation	0.75
Total Responses	19

23. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
For big methods, the documentation does help a lot. However, again, I would improve it grouping the results for "variable". Especially when there are multiple variable tested and each one is tested multiple times.
Too much information
This test case is more complicated than the previous ones. Consequently it has the longest description. It is a little confusing that some of the variables named "itemModel" have a description of how they are obtained but others do not. Inspection of the source code reveals that this is due to some of the variables that have that name are created as a parameter in the assertions as opposed to one that was setup earlier. There are also several very verbose declarations of variables that are not mentioned in the description at all though looking at the actual code this could be because it does not appear that these variables are actually used in any of the validations.
I believe "var model" and "var itemModel" are key variables for this test however I don't think the sentence "This unit test case method tests match constraint message to route." explain why we generate this two variables. Other problems are similar to previous cases.
Overally verbose.
Continuing response from the previous test case (I accidentally moved forward somehow and can't return): the `proxy` and `mixin` variables have their origins described twice, which is redundant. As for this test case, it again has origins of some variables explained multiple times. What I find confusing however (and this may perhaps be due to the fact that I have not used C# before), is that the origin of `itemModel` is not given when usage of its fields is preceded by "List". I would have expected it to do so, so for me this affects the understandability of the description. That being said, since this information can be found elsewhere, I do not find any information to be missing.
Case 5,6, and 10 are missing a slicing path and for the others the slicing paths are the same
Same as all previous comments, description is unclear, slicing path should show the exact code, and variable origins are repeated.
Q1: in "This unit test case includes following focal methods: " part, may add "var itemModel = model[0]; Q2: either add variable description in (5) (6) (10) in "This unit test case validates that" part or delete all variable description part. Q3: easy to understand.
The description helped a lot on reading and understanding the code
For longer unit test methods the description can become cumbersome to read.
Same as before.
It's clear with a little redundancy. It does become a little less expressive when the same variable is tested and the descriptions are separated, but that relates more to the code than the description
easy to read
Same issues as before - code easier to read.
It would be useful to get the types of the variables model and context. In this case, the test case was quite long. But the slicing paths helped me to follow all the lines of code that are relevant for a particular assert. The message is nicely written.
The description captures what is being done but the validation explanations are very long and detailed. The validations steps are only slightly easier to understand as opposed to reading the code.

Statistic	Value
Total Responses	17

24. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		4	21%
3	The description misses the majority of the important information to understand the unit test case.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.32
Variance	0.34
Standard Deviation	0.58
Total Responses	19

25. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		13	68%
2	The description contains some redundant/useless information.		6	32%
3	The description contains a lot of redundant/useless information.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.32
Variance	0.23
Standard Deviation	0.48
Total Responses	19

26. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		15	79%
2	The description is somewhat readable and understandable.		3	16%
3	The description is hard to read and understand.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.26
Variance	0.32
Standard Deviation	0.56
Total Responses	19

27. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
Again, this test case is relatively simple and the description does a good job covering the relevant information while being easy to read.
For Q1, this unit test case validates that result is equal to Response. But we only see where result is obtained from and we missed what is "Response".
A little more verbose than needed.
The description concisely describes what happens and where the relevant variables in the assertion originate.
The information about the slicing path doesn't provide information about how exactly the other variable affect main variable.
Same as all previous comments, no redundant information because there was only one assertion so intermediate variables were only used once.
Q2: variable client and variable schema can combine into one sentence. Q1 and Q3: enough information and easy to understand.
This description is clear enough
The description is accurate and easy to understand.
It captures the dependency of result on the two variables
Easy to read.
Same issues as before - code easier to read.
In this case I could not find the type of the SUT in the provided description. The title of the description suggests that the SUT is the class BaseClientServiceTest. However, by looking at the code one can find that the SUT is in fact the class CreateClientService (i.e. the variable "client" is instanced as a CreateClientService). This is important information that should be included in the description.
Although I don't prefer as many details as they provide. The description is well said and easy to understand for the level of detail.

Statistic	Value
Total Responses	14

28. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		5	26%
3	The description misses the majority of the important information to understand the unit test case.		0	0%
	Total		19	

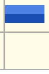
Statistic	Value
Min Value	1
Max Value	2
Mean	1.26
Variance	0.20
Standard Deviation	0.45
Total Responses	19

29. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		16	84%
2	The description contains some redundant/useless information.		3	16%
3	The description contains a lot of redundant/useless information.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.16
Variance	0.14
Standard Deviation	0.37
Total Responses	19

30. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		14	74%
2	The description is somewhat readable and understandable.		3	16%
3	The description is hard to read and understand.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.37
Variance	0.47
Standard Deviation	0.68
Total Responses	19

31. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
When the value of an assertion is a new object (or a combination) the description does not seem to help a lot with the understanding. Like this: (1) uri is equal to new Uri(AuthorizationCodeUrl.ToString() + "?response_type=code")
"?response_type=code" looks weird. Remove "?"
Another good description of a relatively simple unit test. This one did not have the redundancy issues similar ones had. The relevant information is presented in the description of the focal method and is not repeated in the description of the assertions.
This is a simple test and I don't see any distinct deficiency.
In my opinion a description as the following one: This unit test case validates that: (1) uri is equal to new Uri(AuthorizationCodeUrl.ToString() + "?response_type=code") is completely unuseful. Such description is not a conceptual description and it does not add anything with respect what I can read in the source code.
Good.
Same as previous.
It is missing some information about how variable uri is affected
I found it a positive that the assertion contained what the value should be equal to. I also liked that it said the focal method is related to the assertion on line 49 and I was able to view that assertion via the popover.
Q3: may need more explanation what is 'new Uri(AuthorizationCodeUrl.ToString() + "?response_type=code") ", use simple sentence to replace those code Q1 and Q2: enough information and concise.
I also think this one is clear enough.
The description simply follows (instead of translates) the logic of the function. It is no better than reading the code directly due to the function is so simple.
It indicates the variables it involves, but it does mention AuthorizationCodeRequestUrl(AuthorizationCodeUrl);(line 47)
Like the rest.
Same issues as before - code easier to read.
The description is almost complete. Just a single comment: "uri" was created using the object request, so request may be in the slicing path of uri. I'm pointing this out because it would be interesting to know based on the description how the object "uri"is created.
This description was too low level and it may have been easier to just look at the code. The language still has some grammar and comprehension problems.

Statistic	Value
Total Responses	17

32. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		9	47%
2	The description misses some important information to understand the unit test case.		8	42%
3	The description misses the majority of the important information to understand the unit test case.		2	11%
	Total		19	

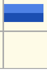
Statistic	Value
Min Value	1
Max Value	3
Mean	1.63
Variance	0.47
Standard Deviation	0.68
Total Responses	19

33. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		15	79%
2	The description contains some redundant/useless information.		4	21%
3	The description contains a lot of redundant/useless information.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.21
Variance	0.18
Standard Deviation	0.42
Total Responses	19

34. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		9	47%
2	The description is somewhat readable and understandable.		8	42%
3	The description is hard to read and understand.		2	11%
	Total		19	

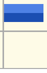
Statistic	Value
Min Value	1
Max Value	3
Mean	1.63
Variance	0.47
Standard Deviation	0.68
Total Responses	19

35. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
This one looks good.
It is a little confusing that the assertions contradict each other, but looking at the focal methods explains this. It could perhaps be useful to say which line the assertions are on, so that we could know that the variable is changing between assertions.
For this test case. I would suggest combining "This unit test case includes following focal methods"(part 1) and "This unit test case validates that"(part 2). In part 2, we generate 3 assertions but they are based on different conditions. Without previous conditions these would look confuse. Actually two conditions are described in part 1 but in part 1 they are related to the code line not the exact assertion in part 2. So I think it's better to combine these two parts.
Also in this case, the summary does not reflect any conceptual information useful for understanding the source code.
The description of validation is not clear.
Overall good, though not perfect.
There is not slicing path for the variable `cols`. The test case still seems fairly straightforward, but I was expecting this based upon the previous test cases.
-It is not considering CollectionAssert. -variable col was modified before each assert
As with all the previous examples, the test description was poor. The assertions did not make complete sense. There should be some additional information like asserts col.count is 0 at first, col.count is 1 after first add, and col.count is 2 after the second add.
Q3: may add line number after col.Count is equal to 0/1/2 to improve readable and understandable. Q1 and Q2: enough information and concise
Q1: It looks like the CollectionAssert.AreEqual method was totally overlooked;
The two adding functions and the assertions for count have dependencies. Before the first adding, the count should be 0; after the first adding, the count increases to 1; after the second adding, the count increases to 2. The description doesn't reveal these dependencies. It is less clear than directly read the code.
The description missed the CollectionAssert's that dealt with color
Does not clearly distinguish between Assert and CollectionAssert - hard to tell that there is an assert on line 45.
Same issues as before - code easier to read.
I would say that this description does a pretty would work showing relevant information about the test case.
The description was comprehensible but the flow of the language still tripped me up. I also think some of the details weren't necessary.

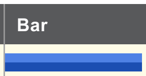
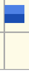
Statistic	Value
Total Responses	17

36. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	74%
2	The description misses some important information to understand the unit test case.		3	16%
3	The description misses the majority of the important information to understand the unit test case.		2	11%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.37
Variance	0.47
Standard Deviation	0.68
Total Responses	19

37. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		7	37%
2	The description contains some redundant/useless information.		11	58%
3	The description contains a lot of redundant/useless information.		1	5%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.68
Variance	0.34
Standard Deviation	0.58
Total Responses	19

38. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		10	53%
2	The description is somewhat readable and understandable.		6	32%
3	The description is hard to read and understand.		3	16%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.63
Variance	0.58
Standard Deviation	0.76
Total Responses	19

39. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
The slicing path is not clear.
There is some redundant information this time as the same variables are used in multiple assertions.
The problem of Q2 we have already discussed in Section 2.
Also in this case, the summary does not reflect any conceptual information useful for understanding the source code. The source code does not contains much lines of code and it can be easily understood without a description as the proposed one.
It doesn't show the focal method.
Too much text.
The origin of \`request\` is given for 5 of the 6 assertions.
-Description is not well written -Doesn't describe the assert -Some of the slicing paths in couple of cases are the same as others
Repeated information on where the variables come from because they are used in multiple assertions.
enough information, simpler and easy to understand.
Q2: The part of how request is formed in (1) to (5) looked like redundant information - appearing only once for the same thing will be enough
The description simply copy the function name, which is hard to understand.
The variable dependencies for all but one of the assertions have the same source, but it's repeated. Again, I subtract this from Q2 rather than Q3 because there are cases where it can change, but in this case, I only need to see the variables uri and query once.
Easy to read.
Same issues as before - code easier to read.
The description does a good work describing the asserts. However, it should show how the sut's are set up. It seems that there are two of them: HttpRequestMessage and MaxUrlLengthInterceptor. I would say that the problem lies in the test case itself and not in the description. It seems that its developers were trying to test several things in a single test case.
The description was not very helpful at all since it was tough to read however they did have a lot of detail probably top much.

Statistic	Value
Total Responses	17

40. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		13	68%
2	The description misses some important information to understand the unit test case.		6	32%
3	The description misses the majority of the important information to understand the unit test case.		0	0%
	Total		19	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.32
Variance	0.23
Standard Deviation	0.48
Total Responses	19

41. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		8	42%
2	The description contains some redundant/useless information.		6	32%
3	The description contains a lot of redundant/useless information.		5	26%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.84
Variance	0.70
Standard Deviation	0.83
Total Responses	19

42. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		6	32%
2	The description is somewhat readable and understandable.		9	47%
3	The description is hard to read and understand.		4	21%
	Total		19	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.89
Variance	0.54
Standard Deviation	0.74
Total Responses	19

43. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
The high-level description is not clear.
As the same variable is updated and used multiple times, this unit test description is very redundant. This also makes it harder to read. Perhaps the focal methods that are the same but used multiple times as a single method can simply be listed once with a list of the lines it appears on. Something similar could be done with the redundant assertions.
Actually only in the last part of this test we look into the ETag. Before that assertion we have many other assertions but in the general description we only see "tests create request e tag". For other part, since there are more than ten assertions, we could see many information are just repeated.
In this case the source code is particularly complex, and, even if the description does not contain any conceptual information, it is able to group together the different responsibilities of the method.
It's not clear what variable body do.
Too much.
In this case, I think it is somewhat redundant that the \`CreateRequest()\` method is listed as a focal method multiple times. That said, the fact that the origin of \`request\` is given multiple times is helpful here given the knowledge that it is created in the same way, but in multiple places. So I still think it is overall redundant, but it is helpful that it is clear the method is used multiple times. Becuase viewing the slicing path is particularly relevant here, I think it is somewhat less readable than previous test cases. Overall, it still has an acceptable level of readabillity.
Most of the test case validations have paths that are in other validations. I think those can be grouped in just one.
It was difficult to tell what each assertion matched to what request. It was also unclear what the difference was with each request.
Q2: can delete "This unit test case includes following focal methods: ", not useful information. Q1 and Q3: enough information and easy to ready and understand.
The core problem of this one is, the httpRequest is taken as the main role - This is right, but knowing it is equal to request.CreateRequest() for several times in a row without knowing what is changed will cause problems on reading understanding, rendering it less effective on showing the purpose and logic of the method and saving our time from reading the codes very carefully.
Again, I think that for long unit test methods, the description becomes difficult to read, perhaps summarizing the assertions for longer methods to give at a glance information would be useful too?
Same as before. It is better to uncover the dependency between the focal methods and assertions.
Here the repetition of the Assertions is necessary because the instances are different. They are listed in order of occurrence, but that should noted automatically for multiple instances of the same check.
Easy to read despite lengthiness
Same issues as before - code easier to read.
The SUT is not properly defined in the description.
I didn't even read the entire description because it was hard to read and would have been easier to just look at the source code. These almost belong as comments as opposed to a concise description.

Statistic	Value
Total Responses	18

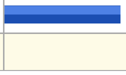
44. General question 1: Do you think identifying of focal methods would help developers to understand the unit test case?

#	Answer	Bar	Response	%
1	Yes		17	89%
2	No		2	11%
	Total		19	

Yes	No
In the case of complicated unit test, identifying automatically the focal method does help the developer.	not so useful, can combine with slicing path
Yes but it's better to combine them with our assertions sentences.	
But only in the cases in which the source code is complex. Otherwise, descriptions is simply superfluous.	
It gives some idea of how the variable works.	
Yes, focal methods is important - although sometimes the related details should also be shown.	
I think helps in understanding of the unit test because it shows assertion dependencies within the method.	
It helps you when looking at the assertions.	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.11
Variance	0.10
Standard Deviation	0.32
Total Responses	19

45. General question 2: Do you think identifying of slicing path would help developers to understand the unit test case?

#	Answer	Bar	Response	%
1	Yes		13	68%
2	No		6	32%
	Total		19	

Yes	No
Yes, but not as displayed in the document.	Just see the line's number won't help me a lot. I personally could find lines of a variable quickly.
As said before, it is absolutely needed when the source code is complex, otherwise only noise is introduced.	Not in the way it is presented - maybe with code highlighting?
yes, very helpful to understand code	
It helps on saving time from reading the codes and finding a small piece from it	
I think this would generally help developers attempting to comprehend unit tests, although sometimes these were a little harder to follow.	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.32
Variance	0.23
Standard Deviation	0.48
Total Responses	19

46. General question 3: Is our generated descriptions useful for understanding the unit test cases in the system?

#	Answer	Bar	Response	%
1	Yes		17	89%
2	No		2	11%
	Total		19	

Yes	No
For complicated unit test cases the description is very useful (even though it becames complicated itself too!), but for simple methods the code looks even simpler than the description	-
Yes.	-
Only in some cases.	-
Somewhat	-
Yes but sometimes the description is a little bit weird grammatically speaking	-
yes, simple and useful	-
Definitely yes. But it also requires the programmers write the code in a better way to keep the description most effective.	-
They were helpful, however, they generally contained only information passed out of the method name, so it seems the method name itself would have been just as useful.	-
Easy to read format.	-

Statistic	Value
Min Value	1
Max Value	2
Mean	1.11
Variance	0.10
Standard Deviation	0.32
Total Responses	19

47. General question 4: What software engineering tasks would you use this type of descriptions for? (Open question)

Text Response
Bug Fixing and software maintenance: when a developer is trying to fix a bug, the first thing he should understand is the test case that fails. The approach can help with that.
It is useful if I am not familiar with an application.
Any software engineering task that used unit testing could benefit from these natural language descriptions.
I think this is useful for us to check the test coverage of an automatically test cases generator.
Big programs.
I think the generated descriptions are overly mechanical and would benefit from a softer, less mechanical approach.
Perhaps the notion of a slicing path could be of use in a bug report if it were clear on what line the bug originated?
At writing or generating new test cases.
If I needed to modify a test case or something else dealing with using the test case. The use seems pretty limited to that.
test task or normal function explanation.
I don't know the area of SE very much. However, I think maybe I can also use this description on build management?
I think these types of descriptions would be really useful in understand unit tests for the purpose of writing/rewriting them for maintenance purposes as code evolves over time. One potentially interesting use for these would be the following: The developer could type in a natural language description of the features he wants tested in the application, and using these natural language descriptions of unit tests, a concise test suite could be generated. This has the potential to greatly improve the software testing cycle.
Detecting Bug or Modifying the previous test cases.
Comment generation or modification.
method descriptions
Sorry, at the moment, I don't believe the summaries are useful.
Software Comprehension. Concept Location. Testing (If one of the tests fails, it would be easier to find the reason)
If I was quickly trying to understand what the code was doing on a high level, then I could delve into the source code with more understanding. Would also be useful to see if some functionality wasn't tested.

Statistic	Value
Total Responses	18

48. General question 5: How could we improve the document? What kind of information would you like to include/remove?

Text Response
I proposed some improvements during the questions. 1) Improve the slicing. The little window with the slicing path does not really help a lot. One could show the source code and augment it highlighting the variables along the path and use some arrows. 2) Group the results by variables. 3) Enable/Disable additional information. Maybe the developer just wants few information while she finds a very busy document page.
The information is good but often redundant. If there was a way to combine repeated information I think it could be much more concise and readable than it already is.
All the suggestions of me have been proposed in Section 2 and 3 with details.
Origins of the variables.
In terms of information, it was good. In terms of ease of being able to read the description at a glance, it is a little lacking. The key there is that it feels mechanical and functional and not quite natural with a hint of elegance.
In all the cases where the origin of a variable was presented in multiple assertions, it might be somewhat better to use more of a table structure. To be more specific, the assertions could be in one column and the origins/slicing paths in another. For those assertions where the origin is the same, you could simply merge the rows in the origin/slicing path column.
Grouping the cases validations that have the same paths will decrease dramatically the text in large examples.
The test descriptions need to be improved. Also showing more code would help.
identifying of focal methods can be removed to make description simpler. identifying of slicing path is a good idea, let others easy to understand and read code.
I think identifying the re-used variables is crucial for a more clear description. Avoid displaying the same thing more than once or show a bit more details around it as a reason. Tables are also welcome.
I think two things could improve the documentation. 1) More readable unit test general descriptions, right now they only contain information parsed form the method name, I think applying more advanced NLP techniques could yield more readable results. 2) Summarize the assertions for larger methods, these methods typically tend to test several related things, and offering a concise summary might help in at-a-glance comprehension.
1. combine the massive validations that have same slicing path or simply show them assertion code(more clear). 2. show dependency among focal methods and assertions.
In the case of x.isEqual(Max + 1), the value could have been replaced with a literal value. In the case of the exact same information being repeated multiple times, show it somewhere on the side and provide a reference to it. Perhaps some of my critiques are more HCI related, but they would have an impact on readability and expressiveness. Showing automatically when a dependency is unique or repeated would be important, otherwise it takes a while to look at the description and the gains are lessened.
Path slicing seemed the least useful - remove?
I don't think it is useful to try to repeat what is in the code without abstracting. Abstraction, however, is difficult, also because the unit tests are often very short and simple code fragments. What is missing (and partly gets lost in the process of summarizing) is information of the context of the test: What are the (direct and indirect) dependencies of the test? Do there exist similar test cases? etc.
Types of variables (for focal methods and slicing paths). Proper definition of SUTs Description of mocked objects (if the test case contains them)
Some of the validation steps should be removed because they seem to get in the way of understanding the test case. Possibly if the validation steps could be explained more rather than short sentences of information.

Statistic	Value
Total Responses	17