

**ASTQB Certified Tester
Test Automation Engineering
Sample Exam Answers**



American Software Testing Qualifications Board

63 points possible. 41 points to pass. 90 minutes.

- 1. (1 pt.) Which of the following is a disadvantage of test automation?**
- a. They are less subject to human error during execution
 - b. They execute faster than manual tests
 - c. They always verify exactly what they were programmed to verify
 - d. They can execute more complex tests more reliably

C is correct. This is a disadvantage because the test coverage can't be expanded by varying the tests. A, B and D are all advantages to using test automation. (CT-TAE-1.1.1 (K2) Explain the advantages and disadvantages of test automation)

- 2. (1 pt.) Which of the following characteristics of Agile software development allows the team to cover all test levels with the appropriate amount and depth of automation?**
- a. The right tools
 - b. The right training for the test automation engineers
 - c. The elimination of silos within the team
 - d. The early publishing of completed requirements

C is correct per the syllabus. Eliminating the silos between developers, testers and other stakeholders helps ensure the proper test automation is built at the right time. A and B are needed for all SDLCs. D is not correct because in Agile the requirements are changing. (CT-TAE-1.2.1 (K2) Explain how test automation is applied across different software development lifecycle models)

3. (1 pt.) You have a highly technical test automation team with all engineers being adept at coding in a wide set of languages. Which of the following would be the best tool to use for implementing test automation for the SUT?

- a. Full code
- b. Commercial off-the-shelf (COTS)
- c. Open-source
- d. A tool based on the same code as the SUT

D is correct. This would be the best choice if possible. This will allow easier integration into the CI/CD pipeline and the developers can also provide some support to the engineers if needed. (CT-TAE-1.2.2 (K2) Outline suitable test tools for a given system under test)

4. (1 pt.) How does observability in the SUT help with test automation development?

- a. a. The TAE can watch the code while it executes
- b. The automation software can access interfaces that will let it determine if the expected results have been achieved
- c. The automation software can access controls such as function calls, to drive the software and verify results
- d. It makes the components and their interfaces more clear and easier to understand

B is correct. By accessing these interfaces, the software can verify if the actual result matches the expected result. (CT-TAE-2.1.1 (K2) Describe the configuration needs of an infrastructure that enable implementation of test automation)

5. (1 pt.) Which environment should be used to run performance testing in order to obtain the most accurate results?

- a. Local dev
- b. Build
- c. Integration
- d. Staging

D is correct. This system should be the closest replication of production. Performance tests can be run in all environments, but until you have everything integrated and a system that is representative of production, the results may not be indicative of production performance. (CT-TAE-2.1.2 (K2) Explain how test automation is leveraged within different environments)

6. (1 pt.) Which environment is best suited for running automated API tests?

- a. Local dev
- b. Build
- c. Integration
- d. Staging

C is correct. The integration environment is the best place to run these automated tests. They can be run earlier, but stubs and mocks will likely be required because the integrations are not in place. (CT-TAE-2.1.2 (K2) Explain how test automation is leveraged within different environments)

- 7. (3 pts.) You are working on a new test automation project. The developers have implemented a CI/CD pipeline with automatic generation of environments as needed. Everyone agrees that it would be best if the test automation code was executed automatically as the code passes through the pipeline. How does this affect your approach to the automation implementation?**
- a. Automated reporting will be critical
 - b. You will need a tool that integrates easily and uses similar technology to the SUT
 - c. Environment setups will need to be tested prior to the execution of any test automation as they are being generated from code
 - d. There will be no need for a test architect since the developers have already implemented CI/CD and the automation code just needs to comply with the pipeline rules

B is correct. In this environment, you will need a tool that allows you to develop code that will work with the SUT code – allowing automated execution within the CI/CD and simplifying the tool landscape. A may be true, but that isn't necessarily indicated by the question. C is less likely to be needed than if the environments were manually configured. D is unlikely because test architects are needed with any test automation solution that is complex. (CT-TAE-2.2.1 (K4) Analyze a system under test to determine the appropriate test automation solution)

8. (3 pts.) You have done the following tool comparison:

	Tool A	Tool B	Tool C	Tool D
Record/Playback capability	O	O	O	X
Integrates with CI/CD pipeline	O	P	P	O
API automation	O	X	P	O
Mobile capability	X	O	O	P
Source	COTS	COTS	Open Source	Open Source

O = out of the box functionality

P = functionality can be implemented with programming

X = functionality is not available

You have assigned the following importance to these areas:

Record/Playback – nice to have

Integrates with CI/CD – nice to have

API automation – must have

Mobile – nice to have

Source – open source preferred

The team who will use this tool has strong programming skills.

Given this information, which is the best tool for your organization?

- a. Tool A
- b. Tool B
- c. Tool C
- d. Tool D

D is the best choice in this case. They don't need the record/playback capability since they are experienced programmers. D is better than C because it has built in API testing capability. (CT-TAE-2.2.2 (K4) Present technical findings of a tool evaluation)

9. (1 pt.) In which part of the test automation design should you find the interface to the test management software?

- a. In the gTAA
- b. In the TAA
- c. In the TAS
- d. In the SUT

A is correct. The interface to the test management software is described in the gTAA. (CT-TAE-3.1.1 (K2) Explain the difference between test automation architecture and solution)

10. (1 pt.) Which is the correct ecosystem in which the test automation runs?

- a. TAF
- b. TAS
- c. TAA
- d. gTAA

B is correct. The test automation runs within the ecosystem of the Test Automation Solution. (CT-TAE-3.1.1 (K2) Explain the difference between test automation architecture and solution)

11. (1 pt.) Which of the following elements of the TAA support connecting to the SUT via APIs?

- a. Test generation
- b. Test definition
- c. Test adaptation
- d. Test execution

C is correct. CT-TAE-3.1.2 (K2) Explain the major capabilities in a test automation architecture

- 12. (2 pts.) You have inherited a set of test automation that was developed over several years by multiple test automation engineers. You have found that the framework is very complicated with the same page and flow models incorporated into multiple scripts. Scripts that access page models for different SUTs are all mixed together in one directory. It also appears that generic user actions have been implemented into multiple scripts. What is the proper pairing between these items and the TAF layers?**
- a. Page and Flow models to the Business Logic Layer, generic user actions to Core Libraries Layer, the remaining script code to Test Scripts Layer
 - b. Page Models and generic user actions to Core Libraries Layer, Flow Models to Business Logic Layer
 - c. Test Scripts to the Test Script Layer, SUT specific code to the Test Adaptation Layer, anything called multiple times to the Core Libraries layer
 - d. Logging and Reporting to the Business Logic Layer, API and Database tests to the Core Libraries, Core Library calls to the Test Script Layer

A is correct. (CT-TAE-3.1.3 (K3) Apply layering of test automation frameworks)

- 13. (2 pts.) You have just started a new job with a company that builds application software. Your manager knows nothing about test automation but likes the idea. He has requested a demo of some test automation on one of their applications in production. He would like to have the demo at the end of this week. You have explained to him that you can put together a quick demo, but that developing real maintainable test automation will take longer. Given this information, what approach would be the most cost effective and expedient to create the demo?**
- a. Capture/playback
 - b. Structured scripting
 - c. Data-driven testing
 - d. Behavior-driven development

A is correct. This will be fastest way to create a demo. The code won't be maintainable, but you've already made that clear to your manager. (CT-TAE-3.1.4 (K3) Apply different approaches for automating test cases)

- 14. (2 pts). You have been maintaining a Test Automation Solution (TAS) and you are finding that you are frequently needing to change the test cases when one of the objects on a page in the UI changes. Because the objects are used by multiple tests, one changed object can result in needing to modify multiple tests. This is taking a considerable amount of time each time the team receives a new code drop from the developers. What design principle should you apply when refactoring the code to make it more maintainable?**
- a. The SOLID principles
 - b. Design patterns
 - c. Page object pattern
 - d. Flow model pattern

C is correct. In this case, capturing all the UI elements in a page model and then having the tests interact with the page model will save considerable maintenance time. In this way, when an element changes on the page, only the model needs to be updated rather than having to update each test script. (CT-TAE-3.1.5 (K3) Apply design principles and design patterns in test automation)

- 15. (2 pts.) You have been assigned to a new test automation project. The developers are using Java with Spring Boot in their deployment pipeline. They would like to see the automation using the same tools, particularly for the deployment and execution of the tests. Given this information, what is a primary consideration you need to have when building the team?**
- a. How many people will you need
 - b. Can you find the people you need with the specific programming skills
 - c. Which test levels will be targeted by the automation
 - d. Where should the team members be located

B is correct. Because there are specific skills required, you need to know if you can find people with those skills. (CT-TAE-4.1.1 (K3) Apply guidelines that support effective test tool pilot and deployment activities)

16. (3 pts.) You have developed a full suite of test automation for a new mobile application. This involves using physical devices for the tests. This has worked well, but you are concerned with future maintenance requirements. Which of the following is an area that should be considered with this approach?

- a. License cost of the devices
- b. Simulator installation responsibility
- c. OS upgrades for the physical devices and retesting/test updates as needed
- d. Acceptance testing using the physical devices versus using simulated devices to gain user approval

C is correct. This is an area to be considered since OSs update frequently and across a large set of physical devices this could take a prohibitive amount of time to install the updates and perform the retesting. This may lead you to conclude that simulators might be better and more flexible. (CT-TAE-4.2.1 (K4) Analyze deployment risks and identify technical issues that can lead to the failure of the test automation project, and plan mitigation strategies)

17. (1 pt.) Which of the following is a maintainability issue with some record/playback tools?

- a. The user interface is difficult to use
- b. The logging is burdensome
- c. The coverage is usually inappropriate
- d. The data is hardcoded

D is a common problem with record/playback tools. They record the inputs and sometimes put them into the code rather than parameterizing these variables. (CT-TAE-4.3.1 (K2) Explain which factors support and affect test automation solution maintainability)

18. (1 pt.) What is the purpose of using static analyzers on test automation code?

- a. They shouldn't be used for test automation code
- b. To ensure a higher quality of the code by checking for coding standards
- c. To optimize the code for faster execution
- d. To find and eliminate any global variables

B is correct. This is what static analyzers do. (CT-TAE-4.3.1 (K2) Explain which factors support and affect test automation solution maintainability)

- 19. (2 pts.) Your organization is using a modern continuous integration system which differentiates between the build and deployment phases of the continuous delivery pipelines. The developers have implemented some testing in the build pipeline and that seems to be working, but you are seeing frequent issues with transaction workflows that are either missing or not implemented correctly. You have now developed tests to cover the problem scenarios. How should this testing be implemented in the pipeline?**
- a. It should be implemented as integration tests in the continuous integration pipeline
 - b. It should be implemented as component integration tests in the continuous deployment pipeline
 - c. It should be implemented as system tests in the continuous deployment pipeline
 - d. It should be implemented as component tests in the continuous integration pipeline

C is correct. These are system level tests and should be implemented in the continuous deployment pipeline. (CT-TAE-5.1.1 (K3) Explain how test automation at different test levels integrates into pipelines)

20. (2 pts.) You are developing performance efficiency tests to combat production issues with slow performance, particularly with any reporting transactions. Where should these be implemented in a CI/CD environment to provide the earliest quality gate?

- a. In the continuous integration pipeline
- b. In the continuous deployment pipeline
- c. After the deployment, in the test environment
- d. After the deployment, in the production environment

B is correct. This would give the earliest feedback possible. A is not correct because reports won't be executable this early in the build cycle. C may be possible, but it is later than would be ideal. D is probably not an acceptable approach because it's very late and it could take down the production environment. (CT-TAE-5.1.1 (K3) Explain how test automation at different test levels integrates into pipelines)

21. (1 pt.) How are test environments influenced by configuration management?

- a. Environments are only accessed through the configuration management software
- b. Environments are secured and security tested when added to configuration management
- c. Environment configuration and access information can be stored in configuration management
- d. Environment access is controlled by the configuration management system and software

C is correct. In the case of multiple environments being used, the configuration and URLs of specific environments can be tracked under configuration management to ensure changes are recorded and versions are clearly saved and reusable. (CT-TAE-5.1.2 (K2) Describe configuration management for testware)

22. (1 pt.) What is the primary purpose of using a feature toggle in testware?

- a. To reduce cost of maintaining the test cases
- b. To allow execution of certain tests depending on the setting of the toggle
- c. To be able to bypass tests that are failing due to a defect
- d. To provide more explicit test reporting information based on the toggle setting

B is correct. The point of a toggle is to determine which test cases or suites will be executed. These toggles usually align with toggles in the SUT. (CT-TAE-5.1.2 (K2) Describe configuration management for testware)

23. (1 pt.) Why is it important to understand the APIs connections?

- a. Developers will be coding according to these requirements
- b. Performance testing is dependent on understanding each API connection
- c. This is where the tester will find the documentation regarding the required parameters
- d. The business logic is controlled through the API connections and their relationships

D is correct. Test automation should ensure it is automating realistic transactions and this requires understanding the business logic that underlies the various APIs and how they relate to each other. (CT-TAE-5.1.3 (K2) Define test automation dependencies for an API infrastructure)

24. (1 pt.) What is the purpose of a contract between services?

- a. It defines how the provider will supply services to the consumer
- b. It defines who has access to which data for security purposes
- c. It defines how a service signs on and off the system
- d. It defines the expected speed of transactions and allows that to be tuned up or down as needed

A is correct. (CT-TAE-5.1.3 (K2) Define test automation dependencies for an API infrastructure)

25. (2 pts.) You have created a test automation suite that is testing critical functionality in a new system. It is finding a significant number of defects. You have repeatedly shown your output logs to your manager, but he is not understanding that most of the test cases are failing. What should you do to enhance your logs to make them more understandable?

- a. Nothing. Get a new manager.
- b. Ensure the start and end of every test is logged and presented in bold font
- c. Add color to the results showing red for failures and green for passes
- d. Add screen shots for all the failures

C is correct. This will help make it obvious to your manager that there is a lot of red on the report. Putting together a summary at the beginning would also be helpful. A might also be a good, if not practical, solution. B should already be done but that won't help your manager understand the logs. D should also already be done, but would help the developers, not your manager. (CT-TAE-6.1.1 (K3) Identify data collection methods from the test automation solution and the system under test)

26. (2 pts.) You have a test automation suite that is encountering a failure. You have investigated the logs, but you can't see where it's failing. You've tried running the tests manually and they work without a problem. What other information should you capture in your scripts to help debug the problem?

- a. Screen shots at the point of the failure
- b. Lines of code executed
- c. Equivalent manual testing hours saved
- d. Number of executions with pass/fail information

A is correct. At this point you want to focus on the one failure that is occurring. Screen shots will likely help. If the point of failure screens aren't enough, you may need to capture more, but at least it will give a good starting point. (CT-TAE-6.1.1 (K3) Identify data collection methods from the test automation solution and the system under test)

27. (3 pts.) You are working on a Customer Relationship Management (CRM) system that uses a third-party address lookup to help the user when they enter an address. This also ensures that only valid addresses are entered into the system. Unfortunately, you are finding that addresses in one state are always failing – Kansas. You can't see any problems in the test automation code, and you have determined that there are no data or formatting errors in the failing addresses. You have tried the same test manually and you get the same failure.

Given this information, what should you do next?

- a. You should run the code under the debugger to be sure there are no errors occurring
- b. You should contact the third-party vendor and log an incident with information regarding the failed addresses
- c. You should capture screen shots of the failure
- d. You should ensure that the addresses being entered are valid addresses

B is correct. You've already made sure that the TAS and SUT are working correctly, and that the data being passed is correct, so it's likely the problem is with the third party. C might be something you do as part of B, but just the screen shots alone wouldn't be helpful. (CT-TAE-6.1.2 (K4) Analyze data from the test automation solution and the system under test)

28. (1 pt.) Who should receive a copy of a test progress report?

- a. Only the testers until the software is ready to release
- b. Management, but only if there is an issue they need to resolve
- c. All stakeholders who have a need and/or interest in the testing
- d. No one – this information should only be tracked in a dashboard that is accessible to the technical team members

C is correct. The test progress report should be published for any stakeholders who have a need or desire to know about the test progress. (CT-TAE-6.1.3 (K2) Explain how a test progress report is constructed and published)

- 29. (2 pts.) You have found that the TAS requires different components depending on the deployment environment. This includes such items as URLs being different, different test data and the use of mocks and stubs. What should you do to make the TAS easily installed in the different environments?**
- a. Ensure all the components are under configuration management with clear versioning and environment requirements
 - b. Create a different TAS for each unique environment and maintain each of them as separate entities to avoid deployment issues
 - c. Provide the developers with links to the different environments so they can deploy the TAS for you in the various environments making whatever changes are required during the deployment effort
 - d. Reduce the use of different test environments and instead just use different versions of the TAS to simulate different environments

A is correct. All components of the TAS should be under configuration management and that includes anything that has to be altered based on environment or even the version of the SUT. B is not correct because this would be a maintenance nightmare as soon as the TAS itself needed to be updated. C is not correct because the developers should not be deploying the TAS – that should be the configuration management team or the test automation team. D is not correct because that doesn't even make sense. (CT-TAE-7.1.1 (K3) Verify the correctness of a test automation environment including test tool setup)

- 30. (2 pts.) You are working on a new TAF, and you are getting some interesting results. When the tests are run under the TAF, they are noticeably slower than when they are run manually. This happens right from the beginning of execution and stays slow throughout. What is a likely cause of this problem?**
- a. There is not enough logging, more should be added for better troubleshooting
 - b. The SUT is not adequately provisioned for the proper performance level
 - c. The data being used by the test automation is making the tests take too long and should be reduced in size
 - d. The TAF is affecting the performance for some reason, more timing information may be needed, but there may be too much logging already occurring

D is correct. It sounds like the TAF is affecting the SUT. Excessive logging is a common cause of issues like this. It's unlikely that it's a memory leak since it's happening right from the start. A is not correct because that may exacerbate the problem. B is not correct because it's OK when the tests are run manually. C is not correct because it's the same system, so if it's a database size issue, the results would be seen with manual or automated tests. (CT-TAE-7.1.1 (K3) Verify the correctness of a test automation environment including test tool setup)

31. (1 pt.) You have been running a set of 10 tests, but you are getting inconsistent results from the tests. What type of “correct” behavior is missing?

- a. Repeatability
- b. Intrusion
- c. Composition
- d. Visibility

A is correct. You are not getting the same results, so these tests are not repeatable. (CT-TAE-7.1.2 (K2) Verify the correct behavior for a given automated test script and/or test suite)

32. (1 pt.) What do you check when you are checking for completeness of a test automation suite?

- a. All tests pass
- b. Approximately 10% of the tests fail consistently
- c. The test data is correct and available
- d. The tests are repeatable across different SUT versions

C is correct per the syllabus. A and B are not feasible. D is more related to repeatability than to composition/completeness. (CT-TAE-7.1.2 (K2) Verify the correct behavior for a given automated test script and/or test suite)

33. (1 pt.) Which of the following would be sufficient to determine that an action has actually occurred?

- a. Add a user and verify that the user can no longer log in
- b. Add a user and then delete that user
- c. Add a user, delete the user, add the user again
- d. Add a user and verify in the database that the user exists

D is the only one that actually validates that the action occurred. The others could be masking false positives caused by other failures (such as the delete actually failing). (CT-TAE-7.1.3 (K2) Identify where test automation may produce false-positive and false-negative test results)

34. (1 pt.) Which of the following is a benefit from using a static analysis tool?

- a. The code will run faster
- b. The code will have extensive logging
- c. The developer will be given suggestions for fixes to problems
- d. The tool will automatically generate defect reports for any found problems

C is correct. You often can turn on defect generation, but that's not a recommended practice as it may flood the defect tracking system with problems that won't be fixed – such as code formatting. (CT-TAE-7.1.4 (K2) Explain how static analysis can aid test automation code quality)

35. (2 pts.) You are having repeated failures with the automation every time a new version of the SUT is released. You have done analysis and discovered that the majority of problems are due to issues with the API response elements changing. What is a cost-effective and time-saving approach that you can implement to deal with this issue?

- a. Implement mocks and stubs for the services being called via the APIs and use those in your test automation
- b. Develop your own APIs to call the services and code your automation to use your own controlled APIs
- c. Implement contracts for the APIs and test via the contracts
- d. Get a schema validation tool that will check that the APIs comply to the requirements

D is correct. That would be the easiest way to quickly check before executing the tests. A would not help because it would mask the problem. B would not be a valid test. C is what should be done, but that has to be initiated by the developers. (CT-TAE-8.1.1 (K3) Identify opportunities for improving test cases through data collection and analysis)

36. (3 pts.) You have been reviewing the test automation suite and have found that at least 40% of the code is reused more than twice. What should you do to improve the maintainability of the test suite?

- a. Delete the duplicated code
- b. Create a library in the framework and store the common code there, then update the individual scripts to call the library code modules as needed
- c. Run a schema validation to ensure the code is accurate and the data is being handled correctly
- d. Use static analysis to ensure the re-used code is clean and efficient

B is correct. Since the code is re-used multiple times, it should be removed from the scripts and put into a library in the framework where it can be called by multiple scripts and maintained only in one place. A is not correct because the code is needed. C is not correct as this is not the problem. There is no indication that the code is not current. D is also not correct although it might be a good idea to run the static analyzer after the code is made into a callable module in the TAF. (CT-TAE-8.1.1 (K3) Identify opportunities for improving test cases through data collection and analysis)

37. (2 pts.) You have a test automation suite for a health care application. This application interfaces with a number of external systems via APIs. You have found that the APIs don't necessarily return the right codes when a request times out and you're finding that your automation is getting stuck as you are checking for specific values. In addition to reporting this as a defect, what should you do to make your automation more robust?

- a. Ensure that there are waits and timeouts coded in your automation to avoid an eternal wait
- b. Stop testing as this is a fundamental defect and invalidates any further tests
- c. Terminate your test when the wait exceeds five seconds as this is adequate time for a response
- d. Wait until you get a response because continuing on would not make sense for the test

A is correct. Anytime waits are used, there should be a timeout to ensure you get out of the wait in the case of an error. B and C might not be warranted as there is nothing to indicate you couldn't continue without the data being retrieved by the call (you might be able to use default values, etc.). D is not valid as this could become an eternal wait. (CT-TAE-8.1.2 (K3) Analyze the technical aspects of a deployed test automation solution and provide recommendations for improvement)

38. (2 pts.) You have a need for your test automation scripts to comb through the SUT system logs looking for specific values. Currently that check is coded into each script because each script is seeking different information. How should this be implemented to reduce maintenance?

- a. The logs should be checked manually because that will be more flexible and will require no maintenance
- b. The values checked should be compiled together so the search will always be for all values
- c. A new reusable function should be developed that can accept data values passed from the test script which will search the log for those specific values
- d. A unique function should be developed for each value so that the precise function can be called by the appropriate test script

C is correct. One function should be developed that can take in variables regarding the values to be searched. (CT-TAE-8.1.2 (K3) Analyze the technical aspects of a deployed test automation solution and provide recommendations for improvement)

39. (2 pts.) One of the test automation engineers has built a function for handling GUI components. This function takes the control ID, control type and the action as input parameters and then it executes the action on the specified control. The TAE says all common controls (buttons, check boxes, text boxes, drop downs, etc.) have already been added, but more controls can be added easily to this function. He also pointed out that it will be available for any test automation script that needs to access a particular control. Is this a good approach?

- a. No, it is better for a function to deal with only one type of control at a time, such as one for buttons and another for drop down lists
- b. No, this function will require considerable maintenance as the GUI adds more pages
- c. Yes, this is an example of a good practice that will put common code into one function that can be maintained in one place but used by many test automation scripts
- d. Yes, this function will be easy to extend as more individual buttons, dropdowns and text boxes are added to the GUI assuming these will all behave in an expected manner for their control type

C is correct. This is a good practice and will improve maintenance and reusability. A is to correct as this approach is fine – there is no need to separate this into multiple modules. B is not correct. If anything, adding more pages to the GUI will just reuse the existing controls so will actually reduce maintenance. D is not correct because there would be no need to extend this function as new instances of the controls are added to the SUT. (CT-TAE-8.1.3 (K3) Analyze the automated testware, to understand where consolidation and updates should be made following a given set of test environment or system under test changes)

40. (1 pt.) You have been asked to develop test automation for a system that has no documented requirements and no existing test cases. You do have access to the business users of the system. How could test automation be used to help you determine what you need to automate?

- a. a. The business users could use the record/playback capability to record what they do, and you could then use those recordings as examples to develop maintainable test automation code.
- b. You could use the tools to prepare test data for the users
- c. A test automation script could be used to configure the test environment
- d. The developers could use the TAS to define their solution and could implement their code in the TAF which would let you develop the test automation solution from the SUT source code

A is correct. This could be done by turning on the record/playback capability to see what the users are doing and then using this information to write good test automation code. B and C can both be done, but that doesn't help you determine what to automate. D is just crazy! (CT-TAE-8.1.4 (K2) Summarize opportunities for use of test automation tools)

Bonus from 3.1.5 (2 pts.) You are reworking a TAS to accommodate changes made to the SUT. You have found that the test automation engineer who wrote most of the test scripts used a record/playback tool to find all the UI components on each page to be tested. The objects were then directly referenced from the scripts that used the components. You noticed this problem when you needed to update the test automation scripts because the “gender” radio buttons had to be changed to a drop-down list. This required changes on multiple pages for this web application.

What design approach should the previous TAE have used to make this code more maintainable?

- a. Flow model pattern
- b. Page object pattern
- c. SOLID principles
- d. Object-oriented programming principles

B is correct. A page object model should be created that will capture all the objects shown on a page and that one model can be referenced by all scripts interacting with the objects. That will mean only the model will have to change if an object changes.

Bonus from 3.1.5 (2 pts.) You are currently using a page object model approach for your test automation for a mobile banking application. This has significantly helped the maintenance effort, but you are still spending time fixing scripts that cover common user transactions such as login and check account balance. What would be the best design approach to reduce the time you are spending coding transactional workflows?

- a. Flow model pattern
- b. Page object pattern
- c. SOLID principles
- d. Object-oriented programming principles

A is correct. You are coding commonly used flows, so by using a flow model pattern that overlays the object page model you already have in place, you should be able to reduce the time you're spending modifying and changing the common flows.