Scrum

PSD-I
Professional Scrum Developer (PSD I)

Questions & Answers PDF

For More Information:
https://www.certswarrior.com/

Features:

➢ 90 Days Free Updates

➢ 30 Days Money Back Guarantee

➢ Instant Download Once Purchased

➢ 24/7 Online Chat Support

➢ Its Latest Version

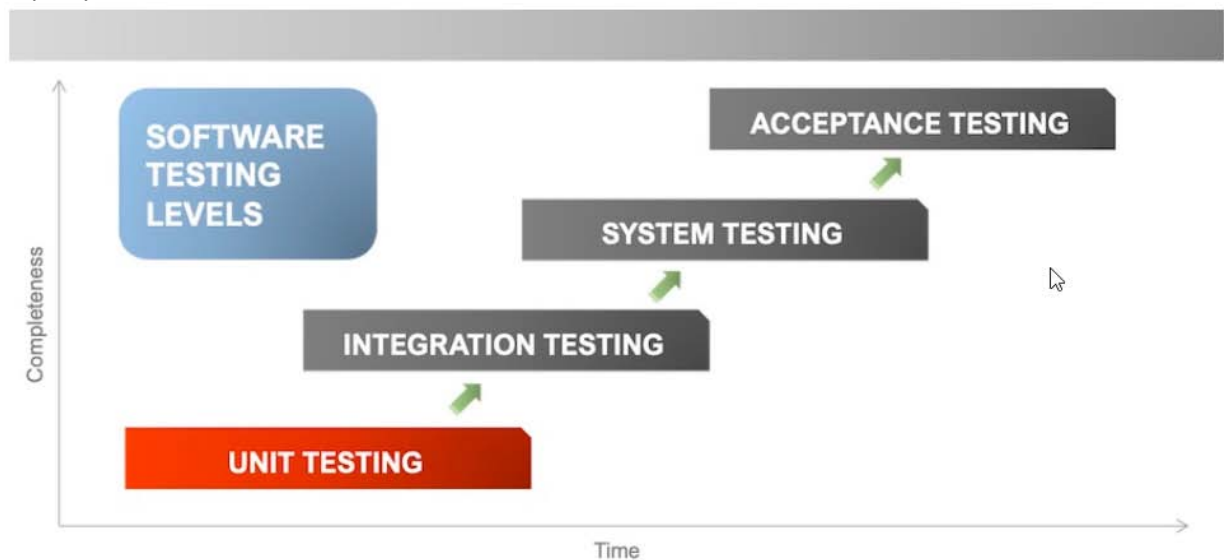# Latest Version: 6.0

Question: 1

A company is trying to reduce the amount of time it puts in manual testing. The new guideline states that the developers should automate the unit test cases, going forward. This is not possible because Unit Testing cannot be Automated.

A. True
B. False

Answer: A

Explanation/Reference:
Unit Test: A Unit test is a way of testing a unit (the smallest piece of code) that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method or property. A Unit test can be automated.

Unit Test is a test that Isolates and Verifies Individual units of source code. Characteristics of a unit test are:
1. Unit test executes fast.
2. Code in each Unit test is as small as possible. Unit test help maintaining readability of the code.
3. Each Unit test is independent of other unit tests.
4. Each Unit test makes assertions about only one logical concept.

Question: 2

A company is trying to reduce the amount of time it puts in manual testing. It does instruct its testing team to automate the regressions test cases going forward. This is not possible because Regression Testing cannot be automated.

A. True
B. False

<div style="border:1px solid black">**Answer: B**</div>

Explanation/Reference:
Regression Test: Whenever developers change or modify their software, even a small tweak can have unexpected consequences. Regression testing is testing existing software applications to make sure that a change or addition hasn't broken any existing functionality. Its purpose is to catch bugs that may have been accidentally introduced into a new build or release candidate, and to ensure that previously eradicated bugs continue to stay dead. By re-running testing scenarios that were originally scripted when known problems were first fixed, you can make sure that any new changes to an application haven't resulted in a regression or caused components that formerly worked to fail. A Regression test can be automated.

Question: 3

John, a developer is discussing with Sam (Product Owner) about the advantages of automating the software build process. The reasons to automate the software build process are? (Choose 3)

A. Automation improves the product by making builds with less errors.
B. Automation accelerates the compile and link processing, thus making more time for feedback.
C. Automating software build eliminates bugs.
D. Code reviews are much faster if you automate your build.
E. Helps find defect and configuration issues quickly.

<div style="border:1px solid black">**Answer: A,B,E**</div>

Explanation/Reference:
Build automation is the process of automating the creation of a software build. It also includes the associated processes such as compiling computer source code into binary code, packaging binary code and running automated tests. The advantages of build automation to software development projects include:
1) Helps find defect and configuration issues.
2) Improve product quality
3) Accelerate the compile and link processing, thus get more feedback.
4) Eliminate redundant tasks
5) Minimize "bad builds"
6) Eliminate dependencies on key personnel
7) Have history of builds and releases in order to investigate issues
8) Improving Product Quality by making builds less error prone. i.e eliminating a source of variation, and thus of defects; a manual build process containing a large number of necessary steps offers as many opportunities to make mistakes

Question: 4

Continuous Delivery makes sure that team is always working towards a "Production Ready" software.

A. False
B. True

Answer: B

Explanation/Reference:
Continuous Delivery is a software delivery practice similar to Continuous Deployment except a human action is required to promote changes into a subsequent environment along the pipeline.
Continuous Delivery is the aim of keeping the system in "Production Ready" state (where it always able to release) to enable the release of a product to the end user on demand.
Continuous Delivery doesn't mean every change is deployed to production ASAP. It means every change is proven to be deployable at any time.
Continuous Delivery helps to move away from the activity of preparing and making software Ready. Instead the Scrum Team work in a way that the software is always "Production Ready".
To support Continuous Delivery, the Scrum Team adds to the Definition of "Done":
1) "Deployed to Production"
2) "Ready to Release"

Question: 5

There is an assigned Role (such as Quality Analyst) who develops and executes the Test Cases.

A. False
B. True

Answer: A

Explanation/Reference:
There is no assigned Role (e.g. QA) who conducts the Test Cases. Developers are responsible for writing and executing the Test Cases.

Question: 6

Branching is:

A. Modifying the main branch or physical code with a version control number so that the code is locked and changed in isolation.
B. Creating a logical or physical copy of code within a version control number so that the copy might be changed in isolation.

Explanation/Reference:
Branching is creating a logical or physical copy of code within a version control number so that the copy might be changed in isolation. Branching also implies the ability to later merge or integrate changes back onto the Parent branch.

Question: 7

The approach to use a branch to work on a feature until it's complete, then merge into the trunk/master branch is called:

A. Task Branching
B. Feature Branching
C. Spin off Branching
D. Release Branching

Answer: B

Explanation/Reference:
Branching is creating a logical or physical copy of code within a version control number so that the copy might be changed in isolation. Branching also implies the ability to later merge or integrate changes back onto the Parent branch.
Branching models often differ between teams and typically depends upon how much work remains in a branch before getting merged back into master. Some of the common type of Branching's / Branching Strategies are:
Release branching: Release branching refers to the idea that a release is contained entirely within a single branch.
Feature branching: Feature Branching is an approach that uses a branch to work on a feature until it's complete. Once the Merge is complete the code is merged back into the trunk/master Branch. Feature branches are often coupled with feature flags–"toggles" that enable or disable a feature within the product. That makes it easy to deploy code into master and control when the feature is activated, making it easy to initially deploy the code well before the feature is exposed to end-users.
Task Branching: Every organization has a natural way to break down work in individual tasks inside an issue tracker, like Jira Software. Issues then becomes the team's central point of contact for that piece of work. Task branching, also known as issue branching, directly connects the issues with the source code. Each issue is implemented on its own branch with the issue key included in the branch name. It's easy to see which code implements which issue by looking at the issue key in the branch name. With that level of transparency, it's easier to apply specific changes to master or any longer running legacy release branch.

Question: 8

While counting the Lines in Lines of Code - Code Matrix it is recommended that the braces, white

space and member declarations are excluded.

A. True
B. False

<div style="border:1px solid black; text-align:center">

Answer: A

</div>

Explanation/Reference:
Lines of Code indicates the number of executable lines of code in a method. This count is an approximate number, based on the Intermediate Language (IL) code.
Lines of Code includes only executable lines of code, so comments, braces, white space and member declarations are excluded.
For Lines of Code the more lines of code in your application, the more code there is to maintain. For Lines of Code, a low value is good, and a high value is bad.
Line of code is not a metrics of code quality. E.g. Codebase A cannot be said to be of worse quality than codebase B on the grounds that its line-count is higher. The scope of work encompassed by codebase A might simply be far greater.

Question: 9

A company is trying to reduce the amount of time it put in manual testing. It does instruct its testing team to automate the exploratory test cases going forward. This is not possible because Exploratory Testing cannot be Automated.

A. False
B. True

<div style="border:1px solid black; text-align:center">

Answer: B

</div>

Explanation/Reference:
Exploratory Test: Exploratory testing is all about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester. It is defined as a type of testing where Test cases are not created in advance, but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a "thinking" activity. Exploratory test cannot be automated.

Question: 10

The benefit of naming standards is that it:

A. Makes the code more readable.
B. Makes the code more communicable
C. Identifies Developers
D. All of the above / Listed Options.

Explanation/Reference:
Scrum teams follow a consistent coding standard so that all the code looks as if it has been written by a single, knowledgeable programmer. The specifics of the standard each team uses are not important; what matters is that the team takes a consistent approach to writing the code. The benefit of establishing naming standards for code is that it makes the code more Readable.

Question: 11

A group of developers are looking at the code they have developed so far. They have a few analysis which show the code coupling matrix across different feature modules. Which of the following is false?

A. When Code Coupling is high, the team may assume that the software modules are less interdependent, i.e., they can function without each other.
B. When code coupling is high, the team may assume that the software modules are interdependent, i.e., they cannot function without the other.

Answer: A

Explanation/Reference:
Coupling is the degree of interdependence between software modules. Coupling is a measure of how closely connected two routines or modules are. It is a measure of the strength of the relationships between modules. In essence, coupling indicates the strength of relation between software modules. When this coupling is high, we may assume that the software modules are interdependent, i.e., they cannot function without the other.

Question: 12

Which of the following statement about DRY is true?

A. Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.
B. Every piece of knowledge must have multiple representations within a system.

Answer: A

Explanation/Reference:
The DRY principle is stated as "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system". The principle has been formulated by Andy Hunt and Dave Thomas in their book The Pragmatic Programmer.
Don't Repeat Yourself (DRY) is a principle of software development which aims at reducing repetition of software patterns, replacing them with abstractions or data normalization. DRY reduces redundancy.

DRY Benefits are:
1) It saves overall development time and effort
2) Code is easy to maintain
3) It reduces the chances of bugs

Question: 13

Which of the following are Good Criteria to include in the Definition of Done?

A. Functional tests passed.
B. Integrated into a clean build.
C. Acceptance criteria met / Acceptance test passed.
D. Unit tests passed.
E. Non-Functional requirements met.
F. Automated regression tests pass.
G. Meets compliance requirements.
H. Feature level functional tests passed.
I. Code reviewed Completed.

Answer: A,B,C,D,E,F,G,I

Explanation/Reference:
The entire Scrum Team must define a Definition of "Done":
1) Developers have the knowledge and skills to do the work to create useable Increments, so they should bring this expertise into the Definition of Done.
2) Product Owners often have inputs related to quality from the business perspective.
3) Scrum Masters helps facilitate improvements to a Definition of Done as part of their accountability for the Scrum Team's effectiveness. A Scrum Master can help create greater transparency for the Scrum Team to identify where quality needs to improve.
The entire Scrum Team would need to work together to change the definition of "Done". If more than one Scrum teams use the same definition of "Done", then those Scrum teams should be involved in changing the Definition of Done as well.
The Developers are required to conform to the Definition of Done. The Scrum Team plans ways to increase product quality by adapting and improving the Definition of "Done" as appropriate. So, Definition of Done changes with time.

A good time to change the Definition of Done is at the Retrospective right before the next Sprint. However, this is not mandatory.
A Good Definition of Done provides:
1) · Guidance on the specific patterns to be implemented in code.
2) Communicates about the Quality Standard needed.
3) Unit tests passed.
4) Code reviewed Completed.
5) Acceptance criteria met / Acceptance test passed.
6) Functional tests passed
7) Non-Functional requirements met.
8) Integrated into a clean build

9) Automated regression tests pass
10) Feature level functional tests passed
11) Meets compliance requirements

Question: 14

What is Class Coupling?

A. Class coupling reflects the number of methods calls in a code.
B. Class Coupling is a measure which reflects the strength of each class & its impact in a code.
C. Class Coupling is a measure which reflects the number of classes in a code.
D. Class coupling is a measure of the dependencies a class has on other classes.

Answer: D

Explanation/Reference:
Class coupling: Class coupling is a measure of dependencies a class has on other classes. This dependency is measured through parameters, method calls and interface implementations.
In Class Coupling, the higher this number, the more likely a change in one class will affect other classes. For class coupling, a low value is good, and a high value is bad.

Question: 15

Code Coverage is a measurement indicating the amount of code that is exercised by tests.

A. True
B. False

Answer: A

CERTSWARRIOR

FULL PRODUCT INCLUDES:

Money Back Guarantee

Instant Download after Purchase

90 Days Free Updates

PDF Format Digital Download

24/7 Live Chat Support

Latest Syllabus Updates

For More Information – Visit link below:

http://www.certswarrior.com

Discount Coupon Code:

CERTSWARRIOR10

We Accept

PayPal