



Software Testing Interview Questions

Free Study Material from
www.OneStopTesting.com

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

WinRunner Interview Questions

1) How you used WinRunner in your project?

Ans. Yes, I have been WinRunner for creating automates scripts for GUI, functional and regression testing of the AUT.

2) Explain WinRunner testing process?

Ans.

WinRunner testing process involves six main stages:

- i. Create GUI Map File so that WinRunner can recognize the GUI objects in the application being tested
- ii. Create test scripts by recording, programming, or a combination of both. While recording tests, insert checkpoints where you want to check the response of the application being tested.
- iii. Debug Test: run tests in Debug mode to make sure they run smoothly
- iv. Run Tests: run tests in Verify mode to test your application.
- v. View Results: determines the success or failure of the tests.
- vi. Report Defects: If a test run fails due to a defect in the application being tested, you can report information about the defect directly from the Test Results window.

3) What is contained in the GUI map?

Ans. WinRunner stores information it learns about a window or object in a GUI Map. When WinRunner runs a test, it uses the GUI map to locate objects. It reads an object's description in the GUI map and then looks for an object with the same properties in the application being tested. Each of these objects in the GUI Map file will be having a logical name and a physical description. There are 2 types of GUI Map files.

- i. Global GUI Map file: a single GUI Map file for the entire application
- ii. GUI Map File per Test: WinRunner automatically creates a GUI Map file for each test created.

4) How does WinRunner recognize objects on the application?

Ans.

WinRunner uses the GUI Map file to recognize objects on the application. When WinRunner runs a test, it uses the GUI map to locate objects. It reads an object's description in the GUI map and then looks for an object with the same properties in the application being tested.

5) Have you created test scripts and what is contained in the test scripts?

Ans.

Yes I have created test scripts. It contains the statement in Mercury Interactive's Test Script Language (TSL). These statements appear as a test script in a test window. You can then enhance your recorded test script, either by typing in additional TSL functions and programming elements or by using WinRunner's visual programming tool, the Function Generator.

6) How does WinRunner evaluates test results?

Ans.

Following each test run, WinRunner displays the results in a report. The report details all the major events that occurred during the run, such as checkpoints, error messages, system messages, or user messages. If mismatches are detected at checkpoints during the test run, you can view the expected results and the actual results from the Test Results window.

7) Have you performed debugging of the scripts?

Ans.

Yes, I have performed debugging of scripts. We can debug the script by executing the script in the debug mode. We can also debug script using the Step, Step Into, Step out functionalities provided by the WinRunner.

8) How do you run your test scripts?

Ans.

We run tests in Verify mode to test your application. Each time WinRunner encounters a checkpoint in the test script, it compares the current data of the application being tested to the expected data captured earlier. If any mismatches are found, WinRunner captures them as actual results.

9) How do you analyze results and report the defects?

Ans.

Following each test run, WinRunner displays the results in a report. The report details all the major events that occurred during the run, such as checkpoints, error messages, system messages, or user messages. If mismatches are detected at checkpoints during the test run, you can view the expected results and the actual results from the Test Results window. If a test run fails due to a defect in the application being tested, you can report information about the defect directly from the Test Results window. This information is sent via e-mail to the quality assurance manager, who tracks the defect until it is fixed.

10) What is the use of Test Director software?

Ans.

TestDirector is Mercury Interactive's software test management tool. It helps quality assurance personnel plan and organize the testing process. With TestDirector you can create a database of manual and automated tests, build test cycles, run tests, and report and track defects. You can also create reports and graphs to help review the progress of planning tests, running tests, and tracking defects before a software release.

11) How you integrated your automated scripts from TestDirector?

Ans When you work with WinRunner, you can choose to save your tests directly to your TestDirector database or while creating a test case in the TestDirector we can specify whether the script is automated or manual. And if it is an automated script then TestDirector will build a skeleton for the script that can be later modified into one which could be used to test the AUT.

12) What are the different modes of recording?

Ans. There are two type of recording in WinRunner.

- i. Context Sensitive recording records the operations you perform on your application by identifying Graphical User Interface (GUI) objects.
- ii. Analog recording records keyboard input, mouse clicks, and the precise x- and y-coordinates traveled by the mouse pointer across the screen.

13) What is the purpose of loading WinRunner Add-Ins?

Ans. Add-Ins are used in WinRunner to load functions specific to the particular add-in to the memory. While creating a script only those functions in the add-in selected will be listed in the function generator and while executing the script only those functions in the loaded add-in will be executed else WinRunner will give an error message saying it does not recognize the function.

14) What are the reasons that WinRunner fails to identify an object on the GUI?

Ans. WinRunner fails to identify an object in a GUI due to various reasons.

- i. The object is not a standard windows object.
- ii. If the browser used is not compatible with the WinRunner version, GUI Map Editor will not be able to learn any of the objects displayed in the browser window.

15) What do you mean by the logical name of the object.

Ans. An object's logical name is determined by its class. In most cases, the logical name is the label that appears on an object.

16) If the object does not have a name then what will be the logical name?

Ans. If the object does not have a name then the logical name could be the attached text.

17) What is the different between GUI map and GUI map files?

Ans. The GUI map is actually the sum of one or more GUI map files. There are two modes for organizing GUI map files.

- i. Global GUI Map file: a single GUI Map file for the entire application
- ii. GUI Map File per Test: WinRunner automatically creates a GUI Map file for each test created. GUI Map file is a file which contains the windows and the objects learned by the WinRunner with its logical name and their physical description.

18) How do you view the contents of the GUI map?

Ans. GUI Map editor displays the content of a GUI Map. We can invoke GUI Map Editor from the Tools Menu in WinRunner. The GUI Map Editor displays the various GUI Map files created and the windows and objects learned in to them with their logical name and physical description.

19) When you create GUI map do you record all the objects of specific objects?

Ans.

If we are learning a window then WinRunner automatically learns all the objects in the window else we will be identifying those objects, which are to be learned in a window, since we will be working with only those objects while creating scripts.

20) What is the purpose of set_window command?

Ans. Set_Window command sets the focus to the specified window. We use this command to set the focus to the required window before executing tests on a particular window.

Syntax: set_window(, time); The logical name is the logical name of the window and time is the time the execution has to wait till it gets the given window into focus.

21) How do you load GUI map?

Ans. We can load a GUI Map by using the GUI_load command.

Syntax: GUI_load();

22) What is the disadvantage of loading the GUI maps through start up scripts?

Ans.1. If we are using a single GUI Map file for the entire AUT then the memory used by the GUI Map may be much high.

2. If there is any change in the object being learned then WinRunner will not be able to recognize the object, as it is not in the GUI Map file loaded in the memory. So we will have to learn the object again and update the GUI File and reload it.

23) How do you unload the GUI map?

Ans. We can use GUI_close to unload a specific GUI Map file or else we call use GUI_close_all command to unload all the GUI Map files loaded in the memory.
Syntax: GUI_close(); or GUI_close_all;

24) What actually happens when you load GUI map?

Ans. When we load a GUI Map file, the information about the windows and the objects with their logical names and physical description are loaded into memory. So when the WinRunner executes a script on a particular window, it can identify the objects using this information loaded in the memory.

25) What is the purpose of the temp GUI map file?

Ans. While recording a script, WinRunner learns objects and windows by itself. This is actually stored into the temporary GUI Map file. We can specify whether we have to load this temporary GUI Map file should be loaded each time in the General Options.

26) What is the extension of gui map file?

Ans.

The extension for a GUI Map file is “.gui”.

27) How do you find an object in an GUI map.

Ans.

The GUI Map Editor is been provided with a Find and Show Buttons.

i. To find a particular object in the GUI Map file in the application, select the object and click the Show window. This blinks the selected object.

ii. To find a particular object in a GUI Map file click the Find button, which gives the option to select the object. When the object is selected, if the object has been learned to the GUI Map file it will be focused in the GUI Map file.

28) What different actions are performed by find and show button?

Ans. 1.To find a particular object in the GUI Map file in the application, select the object and click the Show window. This blinks the selected object.

2.To find a particular object in a GUI Map file click the Find button, which gives the option to select the object. When the object is selected, if the object has been learned to the GUI Map file it will be focused in the GUI Map file.

29) How do you identify which files are loaded in the GUI map?

Ans.The GUI Map Editor has a drop down “GUI File” displaying all the GUI Map files loaded into the memory.

30) How do you modify the logical name or the physical description of the objects in GUI map?

Ans.You can modify the logical name or the physical description of an object in a GUI map file using the GUI Map Editor.

31) When do you feel you need to modify the logical name?

Ans. Changing the logical name of an object is useful when the assigned logical name is not sufficiently descriptive or is too long.

32) When it is appropriate to change physical description?

Ans. Changing the physical description is necessary when the property value of an object changes.

33) How WinRunner handles varying window labels?

Ans. We can handle varying window labels using regular expressions. WinRunner uses two “hidden” properties in order to use regular expression in an object’s physical description. These properties are regexp_label and regexp_MSW_class.

i. The regexp_label property is used for windows only. It operates “behind the scenes” to insert a regular expression into a window’s label description.

ii. The regexp_MSW_class property inserts a regular expression into an object’s MSW_class. It is obligatory for all types of windows and for the object class object.

34) What is the purpose of regexp_label property and regexp_MSW_class property?

Ans. The regexp_label property is used for windows only. It operates “behind the scenes” to insert a regular expression into a window’s label description.

The regexp_MSW_class property inserts a regular expression into an object’s MSW_class. It is obligatory for all types of windows and for the object class object.

35) How do you suppress a regular expression?

Ans. We can suppress the regular expression of a window by replacing the regexp_label property with label property.

36) How do you copy and move objects between different GUI map files?

Ans. We can copy and move objects between different GUI Map files using the GUI Map Editor. The steps to be followed are: i. Choose Tools > GUI Map Editor to open the GUI Map Editor.

ii. Choose View > GUI Files.

iii. Click Expand in the GUI Map Editor. The dialog box expands to display two GUI map files simultaneously.

iv. View a different GUI map file on each side of the dialog box by clicking the file names in the GUI File lists.

World’s Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

- v. In one file, select the objects you want to copy or move. Use the Shift key and/or Control key to select multiple objects. To select all objects in a GUI map file, choose Edit > Select All.
- vi. Click Copy or Move.
- vii. To restore the GUI Map Editor to its original size, click Collapse.

37) How do you select multiple objects during merging the files?

Ans. Use the Shift key and/or Control key to select multiple objects. To select all objects in a GUI map file, choose Edit > Select All.

38) How do you clear a GUI map files?

Ans. We can clear a GUI Map file using the "Clear All" option in the GUI Map Editor.

39) How do you filter the objects in the GUI map?

Ans. GUI Map Editor has a Filter option. This provides for filtering with 3 different types of options.

- i. Logical name displays only objects with the specified logical name.
- ii. Physical description displays only objects matching the specified physical description. Use any substring belonging to the physical description.
- iii. Class displays only objects of the specified class, such as all the push buttons.

40) How do you configure GUI map?

a. When WinRunner learns the description of a GUI object, it does not learn all its properties. Instead, it learns the minimum number of properties to provide a unique identification of the object.

b. Many applications also contain custom GUI objects. A custom object is any object not belonging to one of the standard classes used by WinRunner. These objects are therefore assigned to the generic "object" class.

c. If a custom object is similar to a standard object, you can map it to one of the standard classes. You can also configure the properties WinRunner uses to identify a custom object during Context Sensitive testing. The mapping and the configuration you set are valid only for the current WinRunner session

LoadRunner Interview Questions

1. What is load testing?

Load testing is to test that if the application works fine with the loads that result from large number of simultaneous users, transactions and to determine whether it can handle peak usage periods.

2. What is Performance testing? -

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe. This should be done standalone and then in a multi user environment to determine the effect of multiple transactions on the timing of a single transaction.

3. Did u use LoadRunner? What version? Yes.

Version 7.2.

4. Explain the Load testing process? -

Step 1: Planning the test. Here, we develop a clearly defined test plan to ensure the test scenarios we develop will accomplish load-testing objectives. **Step 2: Creating Vusers.** Here, we create Vuser scripts that contain tasks performed by each Vuser, tasks performed by Vusers as a whole, and tasks measured as transactions. **Step 3: Creating the scenario.** A scenario describes the events that occur during a testing session. It includes a list of machines, scripts, and Vusers that run during the scenario. We create scenarios using LoadRunner Controller. We can create manual scenarios as well as goal-oriented scenarios. In manual scenarios, we define the number of Vusers, the load generator machines, and percentage of Vusers to be assigned to each script. For web tests, we may create a goal-oriented scenario where we define the goal that our test has to achieve. LoadRunner automatically builds a scenario for us.

Step 4: Running the scenario.

We emulate load on the server by instructing multiple Vusers to perform tasks simultaneously. Before the testing, we set the scenario configuration and scheduling. We can run the entire scenario, Vuser groups, or individual Vusers.

Step 5: Monitoring the scenario.

We monitor scenario execution using the LoadRunner online runtime, transaction, system resource, Web resource, Web server resource, Web application server resource, database server resource, network delay, streaming media resource, firewall server resource, ERP server resource, and Java performance monitors. **Step 6: Analyzing test results.** During scenario execution, LoadRunner records the performance of the application under different loads. We use LoadRunner's graphs and reports to analyze the application's performance.

5. When do you do load and performance Testing? -

We perform load testing once we are done with interface (GUI) testing. Modern system architectures are large and complex. Whereas single user testing primarily on functionality and user interface of a system component, application testing focuses on performance and reliability of an entire system. For example, a typical application-testing scenario might depict 1000 users logging in simultaneously to a system. This gives rise to issues such as what is the response time of the system, does it crash, will it go with different software applications and platforms, can it hold so many hundreds and thousands of users, etc. This is when we set do load and performance testing.

6. What are the components of LoadRunner? -

The components of LoadRunner are The Virtual User Generator, Controller, and the Agent process, LoadRunner Analysis and Monitoring, LoadRunner Books Online.

7. What Component of LoadRunner would you use to record a Script? -

The Virtual User Generator (VuGen) component is used to record a script. It enables you to develop Vuser scripts for a variety of application types and communication protocols.

8. What Component of LoadRunner would you use to play Back the script in multi user mode? -

The Controller component is used to playback the script in multi-user mode. This is done during a scenario run where a vuser script is executed by a number of vusers in a group.

9. What is a rendezvous point? -

You insert rendezvous points into Vuser scripts to emulate heavy user load on the server. Rendezvous points instruct Vusers to wait during test execution for multiple Vusers to arrive at a certain point, in order that they may simultaneously perform a task. For example, to emulate peak load on the bank server, you can insert a rendezvous point instructing 100 Vusers to deposit cash into their accounts at the same time.

10. What is a scenario? -

A scenario defines the events that occur during each testing session. For example, a scenario defines and controls the number of users to emulate, the actions to be performed, and the machines on which the virtual users run their emulations.

11. Explain the recording mode for web Vuser script? - We use VuGen to develop a Vuser script by recording a user performing typical business processes on a client application. VuGen creates the script by recording the activity between the client and the server. For example, in web based applications, VuGen monitors the client end of the database and traces all the requests sent to, and received from, the database server. We use VuGen to: Monitor the communication between the application and the server; Generate the required function calls; and Insert the generated function calls into a Vuser script.

12. Why do you create parameters? - Parameters are like script variables. They are used to vary input to the server and to emulate real users. Different sets of data are sent to the server each time the script is run. Better simulate the usage model for more accurate testing from the Controller; one script can emulate many different users on the system.

13. What is correlation? Explain the difference between automatic correlation and manual correlation? - Correlation is used to obtain data which are unique for each run of the script and which are generated by nested queries.

14. **How do you find out where correlation is required? Give few examples from your projects?** - Two ways: First we can scan for correlations, and see the list of values which can be correlated. From this we can pick a value to be correlated. Secondly, we can record two scripts and compare them. We can look up the difference file to see for the values which needed to be correlated. In my project, there was a unique id developed for each customer, it was nothing but Insurance Number, it was generated automatically and it was sequential and this value was unique. I had to correlate this value, in order to avoid errors while running my script. I did using scan for correlation.
15. **Where do you set automatic correlation options?** - Automatic correlation from web point of view can be set in recording options and correlation tab. Here we can enable correlation for the entire script and choose either issue online messages or offline actions, where we can define rules for that correlation. Automatic correlation for database can be done using show output window and scan for correlation and picking the correlate query tab and choose which query value we want to correlate. If we know the specific value to be correlated, we just do create correlation for the value and specify how the value to be created.
16. **What is a function to capture dynamic values in the web Vuser script?**
- Web_reg_save_param function saves dynamic data information to a parameter.
17. **When do you disable log in Virtual User Generator, When do you choose standard and extended logs?** - Once we debug our script and verify that it is functional, we can enable logging for errors only. When we add a script to a scenario, logging is automatically disabled. Standard Log Option: When you select Standard log, it creates a standard log of functions and messages sent during script execution to use for debugging. Disable this option for large load testing scenarios. When you copy a script to a scenario, logging is automatically disabled Extended Log Option: Select extended log to create an extended log, including warnings and other messages. Disable this option for large load testing scenarios. When you copy a script to a scenario, logging is automatically disabled. We can specify which additional information should be added to the extended log using the Extended log options.
- 18.
19. **How do you debug a LoadRunner script?** - VuGen contains two options to help debug Vuser scripts-the Run Step by Step command and breakpoints. The Debug settings in the Options dialog box allow us to determine the extent of the trace to be performed during scenario execution..

20. **How do you write user defined functions in LR? Give me few functions you wrote in your previous project?** - Before we create the User Defined functions we need to create the external library (DLL) with the function. We add this library to VuGen bin directory. Once the library is added then we assign user defined function as a parameter. The function should have the following format: `__declspec(dllexport) char* <function name>(char*, char*)` Examples of user defined functions are as follows: `GetVersion`, `GetCurrentTime`, `GetPlatform` are some of the user defined functions used in my earlier project.
21. **What are the changes you can make in run-time settings?** - The Run Time Settings that we make are: a) **Pacing** - It has iteration count. b) **Log** - Under this we have `Disable Logging` `Standard Log` and c) **Extended Think Time** - In think time we have two options like `Ignore think time` and `Replay think time`. d) **General** - Under general tab we can set the vusers as process or as multithreading and whether each step as a transaction.
22. **How do you perform functional testing under load?** - Functionality under load can be tested by running several Vusers concurrently. By increasing the amount of Vusers, we can determine how much load the server can sustain.
23. **What is Ramp up? How do you set this?** - This option is used to gradually increase the amount of Vusers/load on the server. An initial value is set and a value to wait between intervals can be specified. To set Ramp Up, go to 'Scenario Scheduling Options'
24. **What is the advantage of running the Vuser as thread?** - VuGen provides the facility to use multithreading. This enables more Vusers to be run per generator. If the Vuser is run as a process, the same driver program is loaded into memory for each Vuser, thus taking up a large amount of memory. This limits the number of Vusers that can be run on a single generator. If the Vuser is run as a thread, only one instance of the driver program is loaded into memory for the given number of Vusers (say 100). Each thread shares the memory of the parent driver program, thus enabling more Vusers to be run per generator.
25. **If you want to stop the execution of your script on error, how do you do that?** - The `lr_abort` function aborts the execution of a Vuser script. It instructs the Vuser to stop executing the Actions section, execute the `vuser_end` section and end the execution.

26. **What is the relation between Response Time and Throughput?** - The Throughput graph shows the amount of data in bytes that the Users received from the server in a second. When we compare this with the transaction response time, we will notice that as throughput decreased, the response time also decreased. Similarly, the peak throughput and highest response time would occur approximately at the same time.
27. **Explain the Configuration of your systems?** - The configuration of our systems refers to that of the client machines on which we run the Users. The configuration of any client machine includes its hardware settings, memory, operating system, software applications, development tools, etc. This system component configuration should match with the overall system configuration that would include the network infrastructure, the web server, the database server, and any other components that go with this larger system so as to achieve the load testing objectives.
28. **How do you identify the performance bottlenecks?** - Performance Bottlenecks can be detected by using monitors. These monitors might be application server monitors, web server monitors, database server monitors and network monitors. They help in finding out the troubled area in our scenario which causes increased response time. The measurements made are usually performance response time, throughput, hits/sec, network delay graphs, etc.
29. **If web server, database and Network are all fine where could be the problem?** - The problem could be in the system itself or in the application server or in the code written for the application.
30. **How did you find web server related issues?** - Using Web resource monitors we can find the performance of web servers. Using these monitors we can analyze throughput on the web server, number of hits per second that occurred during scenario, the number of http responses per second, the number of downloaded pages per second.
31. **How did you find database related issues?** - By running "Database" monitor and help of "Data Resource Graph" we can find database related issues. E.g. You can specify the resource you want to measure on before running the controller and then you can see database related issues

32. **How did you plan the Load? What are the Criteria?** - Load test is planned to decide the number of users, what kind of machines we are going to use and from where they are run. It is based on 2 important documents, Task Distribution Diagram and Transaction profile.
33. **What does vuser_init action contain?** - Vuser_init action contains procedures to login to a server.
34. **What does vuser_end action contain?** - Vuser_end section contains log off procedures.
35. **What is think time? How do you change the threshold?** - Think time is the time that a real user waits between actions. Example: When a user receives data from a server, the user may wait several seconds to review the data before responding. This delay is known as the think time. Changing the Threshold: Threshold level is the level below which the recorded think time will be ignored. The default value is five (5) seconds. We can change the think time threshold in the Recording options of the Vugen.
36. **What is the difference between standard log and extended log?** - The standard log sends a subset of functions and messages sent during script execution to a log. The subset depends on the Vuser type Extended log sends a detailed script execution messages to the output log. This is mainly used during debugging when we want information about: Parameter substitution. Data returned by the server. Advanced trace.
37. **Explain the following functions:** - **lr_debug_message** - The lr_debug_message function sends a debug message to the output log when the specified message class is set. **lr_output_message** - The lr_output_message function sends notifications to the Controller Output window and the Vuser log file. **lr_error_message** - The lr_error_message function sends an error message to the LoadRunner Output window. **lrd_stmt** - The lrd_stmt function associates a character string (usually a SQL statement) with a cursor. This function sets a SQL statement to be processed. **lrd_fetch** - The lrd_fetch function fetches the next row from the result set.
38. **Throughput - If the throughput scales upward as time progresses and the number of Vusers increase, this indicates that the bandwidth is sufficient.** If the graph were to remain relatively flat as the number of Vusers increased, it would be reasonable to conclude that the bandwidth is constraining the volume of data delivered.

39. **Types of Goals in Goal-Oriented Scenario** - Load Runner provides you with five different types of goals in a goal oriented scenario:
1. The number of concurrent Vusers
 2. The number of hits per second
 3. The number of transactions per second
 4. The number of pages per minute
40. **Analysis Scenario (Bottlenecks):** In Running Vuser graph correlated with the response time graph you can see that as the number of Vusers increases, the average response time of the check itinerary transaction very gradually increases. In other words, the average response time steadily increases as the load increases. At 56 Vusers, there is a sudden, sharp increase in the average response time. We say that the test *broke the server*. **That is the mean time before failure (MTBF)**. The response time clearly began to degrade when there were more than 56 Vusers running simultaneously.
41. **What is correlation? Explain the difference between automatic correlation and manual correlation?** - Correlation is used to obtain data which are unique for each run of the script and which are generated by nested queries. Correlation provides the value to avoid errors arising out of duplicate values and also optimizing the code (to avoid nested queries). Automatic correlation is where we set some rules for correlation. It can be application server specific. Here values are replaced by data which are created by these rules. In manual correlation, the value we want to correlate is scanned and create correlation is used to correlate.
42. **Where do you set automatic correlation options?** - Automatic correlation from web point of view, can be set in recording options and correlation tab. Here we can enable correlation for the entire script and choose either issue online messages or offline actions, where we can define rules for that correlation. Automatic correlation for database, can be done using show output window and scan for correlation and picking the correlate query tab and choose which query value we want to correlate. If we know the specific value to be correlated, we just do create correlation for the value and specify how the value to be created.

43. **What is a function to capture dynamic values in the web vuser script?**
- Web_reg_save_param function saves dynamic data information to a parameter.

SilkTest Interview Questions

1) What is SilkTest?

Ans. SilkTest is a software testing automation tool developed by Segue Software, Inc.

2) What is the Segue Testing Methodology?

Ans. Segue testing methodology is a six-phase testing process:

1. Plan - Determine the testing strategy and define specific test requirements.
2. Capture - Classify the GUI objects in your application and build a framework for running your tests.
3. Create - Create automated, reusable tests. Use recording and/ or programming to build test scripts written in Segue's 4Test language.
4. Run - Select specific tests and execute them against the AUT.
5. Report - Analyze test results and generate defect reports.
6. Track - Track defects in the AUT and perform regression testing.

3) What is SilkTest Host?

Ans. SilkTest Host is a SilkTest component that manages and executes test scripts. SilkTest Host usually runs on a separate machine different than the machine where AUT (Application Under Test) is running.

4) What is SilkTest Agent?

Ans. SilkTest Agent is a SilkTest component that receives testing commands from the SilkTest Host and interacts with AUT (Application Under Test) directly. SilkTest Agent usually runs on the same machine where AUT is running.

5) What is 4Test?

Ans. 4Test is a test scripting language used by SilkTest to compose test scripts to perform automated tests. 4Test is an object-oriented fourth-generation language. It consists of 3 sets of functionalities:

1. A robust library of object-oriented classes and methods that specify how a testcase can interact with an application's GUI objects.
2. A set of statements, operators and data types that you use to introduce structure and logic to a recorded testcase.
3. A library of built-in functions for performing common support tasks.

6) What is the DOM browser extension?

Ans. Document Object Model (DOM) browser extension is a SilkTest add-on component for testing Web applications. DOM browser extension communicates directly with the Web browser to recognize, categorize and manipulate objects on a Web page. It does this by working with the actual HTML code, rather than relying on the visual pattern recognition techniques currently employed by the Virtual Object (VO) extension.

7) What is the VO browser extension?

Ans. Virtual Object (VO) browser extension is a SilkTest add-on component for testing Web applications. VO browser extension uses sophisticated pattern recognition techniques to identify browser-rendered objects. The VO extension sees Web pages as they appear visually; it does not read or recognize HTML tags in the Web application code. Instead, the VO extension sees the objects in a Web page; for example, links, tables, images and compound controls the way that you do, regardless of the technology behind them.

8) Is there any problem in using scripts created on v6.0 to 6.5 or higher versions?

Ans. Moving from lower to higher version should not be a problem.... This is a general statement and cannot be true at all instances. I faced problems with scripts working in 6.5 not running in 7.0 because some of the recognition patterns used each changed. And in some situations, finally landed two paths of the script to perform same action based on version. PS: Did not encounter any problems from 6.0 to 6.5.

9) What is SilkTest project?

Ans. A SilkTest project is a collection of files that contains required information about a test project.

10) How to create a new SilkTest project?

Ans.

1. Run SilkTest.
2. Select Basic Workflow bar.
3. Click Open Project on the Workflow bar.
4. Select New Project.
5. Double click Create Project icon in the New Project dialog box
6. One the Create Project dialog box, enter your project name, and your project description.
7. Click OK.
8. SilkTest will create a new subdirectory under SilkTest project directory, and save all files related to the new project under that subdirectory.

11) Is There any function for word count of web page is available in the SilkTest?

Ans. You can use Clipboard functions. Get All the contents by Ctrl+a & ctrl+C. Then parse the List of string ...

12) What is a SilkTest Testplan?

Ans. The SilkTest testplan is an outline that provides a framework for the software testing process and serves as the point of control for organizing and managing your test requirements. A testplan consists of two distinct parts: an outline, which is a formatted description of the test requirements, and statements, which are used to connect the testplan to SilkTest scripts and testcases that implement the test requirements.

13) What's the best way to create a new test script?

Ans. In Automation; creating test scripts involves basically two step:

1. Creating Test Data for all test cases.
2. Writing scripts for automating the test cases.

The first steps, is the outcome of analysis of what all data is required for all of your testcases to be executed. These all should be collected and scripts to be written so that TEST DATA is ready.

The second steps, is basically the test case execution steps automation.

14) How to get button caption?

Ans. Use GetCaption () method.....

14)How to close unexpected window?

Ans. If Order Request window is a popup.

Following code closes the active browser window if it is a popup:

```
[ - ] !(if Browser.ComboBox("#1").exists())  
[ ] Browser.SetActive()  
[ ] Browser.typekeys("")
```

15)What are the types of text lines in a testplan file?

Ans. A testplan file contains text lines. There are 5 types of text lines in a testplan file:

1. Comment - Marked in green color: Providing commentary information.
2. Group description - Marked in black color: Providing descriptions for groups of tests. Tests in a testplan can be grouped into multiple levels of groups.
3. Test description - Marked in blue color: Providing descriptions for individual test.
4. Testplan statement - Marked in dark red color: Providing relations to link scripts, testcases, test data, closed sub testplans or an include file to the testplan.
5. Testplan statement - Marked in dark red color: Providing relations to link scripts, testcases, test data, closed sub testplans or an include file to the testplan.
6. Open subplan file marker - Marked in magenda color: Providing relations to link sub testplans to be included in a master testplan.

16)What are the default testplan attributes?

Ans. SilkTest offers you 3 predefined default attributes:

1. Category: The type of testcase or group of testcases. For example, you can use this attributes to categorize your test groups as "Boundary value tests", "Navigation tests", etc.

2. Component: The name of the application modules to be tested.
3. Developer: The name of the QA engineer assigned to develop the testcase or group of testcases.

17) How to maintain recovery system?

Ans. In your TestCaseExit () function you can incorporate like

If condition == 1

do this

If condition == 2

do that

18) How to define new testplan attributes?

Ans.

1. Make sure your test project is open.
2. Click Testplan/Define Attributes menu. The Define Attributes dialog box shows up. You should see 3 predefined default attributes: Category, Component, and Developer.
3. Click the New button. The New Attribute dialog box shows up.
4. Enter a name for your new attribute. For example: "Level" to indicate the complexity level of test cases.
5. Select an attribute type: Normal, Edit, or Set.
6. Click OK.

19)How to assign attribute values to test cases?

Ans.

1. Make sure your testplan is open.
2. Click on the test case for which you want to assign an attribute value.
3. Click Testplan/Detail menu. The Testplan Details dialog box shows up.
4. Click the Test Attribute tab.
5. Click the Component field. The dropdown list shows up with all values of "Component".
6. Select one of the values in the dropdown list.
7. Click OK.

20) How to assign attribute values to test cases?

Ans.

1. Make sure your Web browser is active and showing your Web application home page. Do not minimize this Web page window.
2. Make sure your test project is open.
3. Click File/New menu. The New dialog box shows up.
4. Select the Test Frame radio button.
5. Click OK. The New Test Frame dialog box shows up with a list all active Web applications.
6. Select your Web application.
7. Enter a test frame name. For example: HomeFrame.inc.
8. Review the window name. It should be the HTML title your Web application. You can rename it, if needed.
9. Click OK to close the New Test Frame dialog box.
10. Click File/Save menu. HomeFrame.inc.

21) What is stored in a test frame?

Ans. A test frame is a text file, which records the following types of information for a Web application:

1. Comment: Commentary information.
2. wMainWindow: A string constant to identify your application's home page.
3. Home page window: An object of class BrowserChild window that holds application home page.
4. sLocation: The URL of the your application's home page.
5. sUserName and dPassword: User name and password if needed to login to your Web application.
6. BrowserSize: A pair of values to indicate the size of the browser window.
7. Home page objects: A list of all objects on the home page, such as HtmlImage, HtmlText, HtmlLinks, etc.

22) What is the syntax of UI object identifier used by DOM extension?

Ans. The DOM browser extension uses the following syntax for Web UI objects:

```
Browser.BrowserChild("page_title").html_class("object_tag")
```

1. "page_title" is the title of the Web page, defined by the HTML
2. "object_tag" is the label of the HTML element. How a HTML element is labeled depending on the type of HTML element.

23) How to add objects of other pages to a test frame?

Ans. If your Web application has pages other than the home page, you should also record their page objects into the test frame:

1. Make sure your Web browser is active and showing another page of your Web application.
2. Make sure SilkTest is running.
3. Click File/Open menu.
4. Select your test frame file. For example: HomeFrame.inc
5. Click OK to open the test frame.
6. Click Record/Window Declarations menu. The Record Window Declarations dialog box shows up.
7. Click your Web application window. Web page objects are recorded in the Record Window Declarations dialog box.
8. Press Ctrl+Alt to pause the recording.
9. Click "Paste to Editor" button. All recorded objects will be inserted into the test frame.
10. Repeat this for other Web pages, if needed.

24) What is DefaultBaseState?

Ans. The DefaultBaseState is a starting point of test project from which the Recovery System can automatically restart your test cases when test cases fail to continue.

25) What is the standard flow of execution of a test case?

Ans.

1. Starting from the base state.
2. Drive the application to the state where the expected result should occur.
3. Verify the actual result against the expected result.
4. Declare the test case as passed or failed.
5. Return to the base state.

26) How to record a test case?

Ans.

1. Run SilkTest.
2. Click Option/Runtime menu. The Runtime Options dialog box shows up.
3. Edit the Use Files field to include your test frame file and the explorer.inc file.
For example: ...\\HomeFrame.inc,extend\\explorer.inc.
4. Make sure IE 5.x DOM is selected.
5. Click OK to close the Runtime Options dialog box.
6. Open your test project.
7. Click Record/Testcase menu. The Record Testcase dialog box shows up.
8. Name your test case. For example: LoginTest.
9. Select DefaultBaseState in the Application State dropdown list.
10. Click Start Recording button. The Record Testcase dialog closes. Your Web application will be automatically started by SilkTest, based on the information in test frame file. SilkTest Editor window closes. The Record Status dialog box shows up.
11. Continue to use your Web application. SilkTest records everything you did on your application.
12. Click the "Done" button on the Recording Status dialog box to stop recording. The Recording Status dialog box closes. The Record Testcase dialog box shows up again.
13. Click Paste to Editor. SilkTest will insert the recorded activities as 4Test statements into a script file. The Record Testcase dialog closes.
14. Click File/Save menu to save the script file. You can enter a script file name. For example, LoginTest.t.

27) How to write a Masterplan?

Ans. You can run all your sub-plans using master plan approach. You just open new plan file and call your subplan/testcase from this file. If you want to run subplan from master plan then syntax is like include: mysubplan.pln

28) How to define an object verification in a test case?

Ans. While recording a test case, you can define verification points to verify UI objects:

1. Make sure you are in the process of recording a testcase.
2. Make sure the Record Status dialog box is on the screen.
3. Make sure your recording reached the Web page that has the UI object you want to verify.
4. Click the background (blank area) of the Web page. Do not click any objects on the page.
5. Press Ctrl-Alt. The Verify Window dialog box shows up. All the objects on the current Web page are listed on the Verify Window dialog box.
6. Select the object to be verified in the object list. Un-select all other objects.
7. Select the property to be verified in the property list. Un-select all other properties.
8. Click OK to close the Verify Window dialog box.
9. Continue your recording.

29) How to run a test case from a test script file?

Ans. A test script file can store multiple test cases. You can run a testcase from a test script file:

1. Open the test script file.
2. Select the test case in the test file.
3. Click Run/Testcase menu. The Run Testcase dialog box shows up.
4. Click the Run button. SilkTest starts to run the test case.
5. Do not touch mouse or keyboard, to avoid interrupting the test case execution.
6. SilkTest finishes executing the testcase. The Results window shows up with the execution result.
7. Review the execution result.

30) What's in the test result file?

Ans. Result summary: The name of the script file. The name of the testcase. The machine on which the tests ran. The starting time and the total elapsed time. The number and percentage of testcases that passed and failed. The total number of errors and warnings.

1. Result detail: List of errors and detailed information.

31) How to link an error in the result file to the script file?

Ans. Make sure the Result window is open with result file.

1. Locate the error message in the result file.
2. Select the error message.
3. Click the Results/Goto Source menu. The original script file opens up showing the place where the error was originated.

32) How to link an error in the result file to the script file?

Ans.

1. Make sure the Result window is open with result file.
2. Click Results/Pass/Fail Report. The Pass/Fail Report dialog box shows up.
3. Select an attribute on which you want the report to be based on. For example: Component.
4. Click the Generate button.
5. SilkTest generates a report in the Pass/Fail Report dialog box.
6. You can print or export the report.
7. Click the Close button to close the Pass/Fail Report dialog box.

33) What is DBTester?

Ans. DBTester is a testing tool that allows you to access a database server directly through ODBC drivers. If your application is a database driven application, you can perform a test through the application UI, and verify data changes in the database with DBTester without using the application UI.

34) What are the functions offered by DBTester?

Ans. DBTester offers 6 functions. You can use them directly in your test cases:

1. DB_Connect: Opens a database connection linking the data through the specified ODBC DSN name. DB_Connect returns a connection handle which can be used on other DBTester functions. SQL statements can be submitted to the database. For example: con = DB_Connect("dsn=dsn_name")
2. DB_Disconnect: Closes the database connection represented by the specified connection handle. All resources related to this connect are also released. For example: DB_Disconnect(con)
3. DB_ExecuteSql: Sends the specified SQL statement to the specified database connection for execution. DB_ExecuteSql returns a query result handler which can be used by the DB_FetchNext function. For example: res = DB_ExecuteSql(con, "SELECT * FROM ...")

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

4. DB_FetchNext: Retrieves the next row from the specified query result handler. For example: DB_FetchNext(res, col1, col2, col3, ...)
5. DB_FetchPrevious: Retrieves the previous row from the specified query result handler.
6. DB_FinishSql: Closes the specified query result handler. For example: DB_FinishSql(res)

35) How to create group and sub group descriptions in a testplan?

Ans. In a testplan, each text line starting from column 0 represents a top level group description. To create sub group description:

1. Move the cursor the next line below the top level group description.
2. Click Outline/Move Right.
3. The text line will be indented to the right to be come a sub group description.

36) How to define values for a testplan attribute?

Ans. You must define values for a testplan before using it:

1. Make sure your test project is open.
2. Click Testplan/Define Attributes menu. The Define Attributes dialog box shows up. You should see 3 predefined default attributes and other attributes defined by yourself.
3. Select an attribute. For example, "Component". The Values box should be empty.
4. Enter a value in Add box. For example, "Catalog".
5. Click Add. Value "Catalog" should be inserted into the Values box.
6. Repeat the last two steps to add more values.

37) Where are the testplan attributes stored?

Ans. Testplan attributes are stored in the testplan initialization file, testplan.ini, in SilkTest installation directory.

38) What is a test frame?

Ans. A test frame is a file that contains information about the application you are testing. Information stored in a test frame will be used as references when SilkTest records and executes testcases. A test frame is stored in an include file with file extension .inc.

39) How to specify a browser extension to a Web application?

Ans.

1. Run SilkTest.
2. Open Internet Explorer (IE).
3. Enter the URL of the Web application.
4. Leave the IE window with the Web application. Don't minimize the IE window.
5. Go back to SilkTest window.
6. Select Basic Workflow bar.
7. Click Enable Extensions on the Workflow bar.
8. The Enable Extensions dialog will show up. Your Web application running in the IE window will be listed in the dialog box.
9. Select your Web application and click Select.
10. The Extension Settings dialog will show up. Click OK to enable the DOM browser extension.

40) How to test your DefaultBaseState?

Ans.

1. Close your Web application and other Web browsers.
2. Make sure your test frame is open.
3. Click Run/Application State menu. The Run Application State dialog box shows up with a list of states. One of them should be DefaultBaseState.
4. Select DefaultBaseState.
5. Click Run button. The Runtime Status dialog box shows up. And the Results File dialog box shows up too.
6. You should see no error message in the results file.

TestDirector Interview Question

1. Types of vies in Datastage Director?
2. Orchestrate Vs Datastage Parallel Extender?
3. What is an Exception ? What are types of Exception ?
4. What are Routines and where/how are they written and have you written any routines before?
5. What are the command line functions that import and export the DS jobs?
6. How many types of database triggers can be specified on a table ? What are they ?
7. What is NLS in datastage? how we use NLS in Datastage ? what advantages in that ?
at the time of ins. ..
8. What are types of Hashed File?
9. What are the datatypes a available in PL/SQL ?

General Testing Interview Questions

1. What is 'Software Quality Assurance'?

Software QA involves the entire software development Process - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention'. (See the Books section for a list of useful books on Software Quality Assurance.)

2. What is 'Software Testing'?

Testing involves operation of a system or application under controlled conditions and evaluating the results (eg, 'if the user is in interface A of the application while using hardware B, and does C, then D should happen'). The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they shouldn't or things don't happen when they should. It is oriented to 'detection'.

Organizations vary considerably in how they assign responsibility for QA and testing. Sometimes they're the combined responsibility of one group or individual. Also common are project teams that include a mix of testers and developers who work closely together, with overall QA processes monitored by project managers. It will depend on what best fits an organization's size and business structure.

3. What are some recent major computer system failures caused by software bugs?

- Media reports in January of 2005 detailed severe problems with a \$170 million high-profile U.S. government IT systems project. Software testing was one of the five major problem areas according to a report of the commission reviewing the project. Studies were under way to determine which, if any, portions of the project could be salvaged.

* In July 2004 newspapers reported that a new government welfare management system in Canada costing several hundred million dollars was unable to handle a simple benefits rate increase after being put into live operation.

* Millions of bank accounts were impacted by errors due to installation of inadequately tested software code in the transaction processing system of a major North American bank, according to mid-2004 news reports. Articles about the incident stated that it took two weeks to fix all the resulting errors, that additional problems resulted when the incident drew a large number of e-mail phishing attacks against the bank's customers, and that the total cost of the incident could exceed \$100 million.

* A bug in site management software utilized by companies with a significant percentage of worldwide web traffic was reported in May of 2004. The bug resulted in performance problems for many of the sites simultaneously and required disabling of the software until the bug was fixed.

* According to news reports in April of 2004, a software bug was determined to be a major contributor to the 2003 Northeast blackout, the worst power system failure in North American history. The failure involved loss of electrical power to 50 million customers, forced shutdown of 100 power plants, and economic losses estimated at \$6 billion. The bug was reportedly in one utility company's vendor-supplied power monitoring and management system, which was unable to correctly handle and report on an unusual confluence of initially localized events. The error was found and corrected after examining millions of lines of code.

* In early 2004, news reports revealed the intentional use of a software bug as a counter-espionage tool. According to the report, in the early 1980's one nation surreptitiously allowed a hostile nation's espionage service to steal a version of sophisticated industrial software that had intentionally-added flaws. This eventually resulted in major industrial disruption in the country that used the stolen flawed software.

* A major U.S. retailer was reportedly hit with a large government fine in October of 2003 due to web site errors that enabled customers to view one another's online orders.

* News stories in the fall of 2003 stated that a manufacturing company recalled all their transportation products in order to fix a software problem causing instability in certain circumstances. The company found and reported the bug itself and initiated the recall procedure in which a software upgrade fixed the problems.

* In January of 2001 newspapers reported that a major European railroad was hit by the aftereffects of the Y2K bug. The company found that many of their newer trains would not run due to their inability to recognize the date '31/12/2000'; the trains were started by altering the control system's date settings.

* News reports in September of 2000 told of a software vendor settling a lawsuit with a large mortgage lender; the vendor had reportedly delivered an online mortgage processing system that did not meet specifications, was delivered late, and didn't work.

* In early 2000, major problems were reported with a new computer system in a large suburban U.S. public school district with 100,000+ students; problems included 10,000 erroneous report cards and students left stranded by failed class registration systems; the district's CIO was fired. The school district decided to reinstate it's original 25-year old system for at least a year until the bugs were worked out of the new system by the software vendors.

* In October of 1999 the \$125 million NASA Mars Climate Orbiter spacecraft was believed to be lost in space due to a simple data conversion error. It was determined that spacecraft software used certain data in English units that should have been in metric units. Among other tasks, the orbiter was to serve as a communications relay for the Mars Polar Lander mission, which failed for unknown reasons in December 1999. Several investigating panels were convened to determine the process failures that allowed the error to go undetected.

* Bugs in software supporting a large commercial high-speed data network affected 70,000 business customers over a period of 8 days in August of 1999. Among those affected was the electronic trading system of the largest U.S. futures exchange, which was shut down for most of a week as a result of the outages.

* January 1998 news reports told of software problems at a major U.S. telecommunications company that resulted in no charges for long distance calls for a month for 400,000 customers. The problem went undetected until customers called up with questions about their bills.

4. Why is it often hard for management to get serious about quality assurance?

- Solving problems is a high-visibility process; preventing problems is low-visibility. This is illustrated by an old parable: In ancient China there was a family of healers, one of whom was known throughout the land and employed as a physician to a great lord.

5. Why does software have bugs?

* Miscommunication or no communication - as to specifics of what an application should or shouldn't do (the application's requirements).

* Software complexity - the complexity of current software applications can be difficult to comprehend for anyone without experience in modern-day software development. Multi-tiered applications, client-server and distributed applications, data communications, enormous relational databases, and sheer size of applications have all contributed to the exponential growth in software/system complexity.

* Programming errors - programmers, like anyone else, can make mistakes.

* Changing requirements (whether documented or undocumented) - the end-user may not understand the effects of changes, or may understand and request them anyway - redesign, rescheduling of engineers, effects on other projects, work already completed that may have to be redone or thrown out, hardware requirements that may be affected, etc. If there are many minor changes or any major changes, known and unknown dependencies among parts of the project are likely to interact and cause problems, and the complexity of coordinating changes may result in errors. Enthusiasm of engineering staff may be affected. In some fast-changing business environments, continuously modified requirements may be a fact of life. In this case, management must understand the resulting risks, and QA and test engineers must adapt and plan for continuous extensive testing to keep the inevitable bugs from running out of control - see 'What can be done if requirements are changing continuously?' in Part 2 of the FAQ. Also see information about 'agile' approaches such as XP, also in Part 2 of the FAQ.

* Time pressures - scheduling of software projects is difficult at best, often requiring a lot of guesswork. When deadlines loom and the crunch comes, mistakes will be made.

•

* egos - people prefer to say things like:

* * 'no problem'

* * 'piece of cake'

* * 'I can whip that out in a few hours'

* * 'it should be easy to update that old code'

* instead of:

* * 'that adds a lot of complexity and we could end up making a lot of mistakes'

* * 'we have no idea if we can do that; we'll wing it'

* * 'I can't estimate how long it will take, until I take a close look at it'

* * 'we can't figure out what that old spaghetti code did in the first place'

If there are too many unrealistic 'no problem's', the result is bugs.

* Poorly documented code - it's tough to maintain and modify code that is badly written or poorly documented; the result is bugs. In many organizations management provides no incentive for programmers to document their code or write clear, understandable, maintainable code. In fact, it's usually the opposite: they get points mostly for quickly turning out code, and there's job security if nobody else can understand it ('if it was hard to write, it should be hard to read').

* Software development tools - visual tools, class libraries, compilers, scripting tools, etc. often introduce their own bugs or are poorly documented, resulting in added bugs.

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

6.How can new Software QA processes be introduced in an existing organization?

- * A lot depends on the size of the organization and the risks involved. For large organizations with high-risk (in terms of lives or property) projects, serious management buy-in is required and a formalized QA process is necessary.
- * Where the risk is lower, management and organizational buy-in and QA implementation may be a slower, step-at-a-time process. QA processes should be balanced with productivity so as to keep bureaucracy from getting out of hand.
- * For small groups or projects, a more ad-hoc process may be appropriate, depending on the type of customers and projects. A lot will depend on team leads or managers, feedback to developers, and ensuring adequate communications among customers, managers, developers, and testers.
- * The most value for effort will often be in (a) requirements management processes, with a goal of clear, complete, testable requirement specifications embodied in requirements or design documentation, or in 'agile'-type environments extensive continuous coordination with end-users, (b) design inspections and code inspections, and (c) post-mortems/retrospectives.

7.What is verification? validation?

- * Verification typically involves reviews and meetings to evaluate documents, plans, code, requirements, and specifications. This can be done with checklists, issues lists, walkthroughs, and inspection meetings. Validation typically involves actual testing and takes place after verifications are completed. The term 'IV & V' refers to Independent Verification and Validation.

8.What is a 'walkthrough'?

- * A 'walkthrough' is an informal meeting for evaluation or informational purposes. Little or no preparation is usually required.

9. What's an 'inspection'?

* An inspection is more formalized than a 'walkthrough', typically with 3-8 people including a moderator, reader, and a recorder to take notes. The subject of the inspection is typically a document such as a requirements spec or a test plan, and the purpose is to find problems and see what's missing, not to fix anything. Attendees should prepare for this type of meeting by reading thru the document; most problems will be found during this preparation. The result of the inspection meeting should be a written report.

10. What kinds of testing should be considered?

* Black box testing - not based on any knowledge of internal design or code. Tests are based on requirements and functionality.

* White box testing - based on knowledge of the internal logic of an application's code. Tests are based on coverage of code statements, branches, paths, conditions.

* Unit testing - the most 'micro' scale of testing; to test particular functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses.

* Incremental integration testing - continuous testing of an application as new functionality is added; requires that various aspects of an application's functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed; done by programmers or by testers.

* Integration testing - testing of combined parts of an application to determine if they function together correctly. The 'parts' can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

* Functional testing - black-box type testing geared to functional requirements of an application; this type of testing should be done by testers. This doesn't mean that the programmers shouldn't check that their code works before releasing it (which of course applies to any stage of testing.)

* System testing - black-box type testing that is based on overall requirements specifications; covers all combined parts of a system.

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

* End-to-end testing - similar to system testing; the 'macro' end of the test scale; involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

* Sanity testing or smoke testing - typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort. For example, if the new software is crashing systems every 5 minutes, bogging down systems to a crawl, or corrupting databases, the software may not be in a 'sane' enough condition to warrant further testing in its current state.

* Regression testing - re-testing after fixes or modifications of the software or its environment. It can be difficult to determine how much re-testing is needed, especially near the end of the development cycle. Automated testing tools can be especially useful for this type of testing.

* Acceptance testing - final testing based on specifications of the end-user or customer, or based on use by end-users/customers over some limited period of time.

* Load testing - testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

* Stress testing - term often used interchangeably with 'load' and 'performance' testing. Also used to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, large complex queries to a database system, etc.

* Performance testing - term often used interchangeably with 'stress' and 'load' testing. Ideally 'performance' testing (and any other 'type' of testing) is defined in requirements documentation or QA or Test Plans.

* Usability testing - testing for 'user-friendliness'. Clearly this is subjective, and will depend on the targeted end-user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers.

* Install/uninstall testing - testing of full, partial, or upgrade install/uninstall processes.

* Recovery testing - testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>

- * Failover testing - typically used interchangeably with 'recovery testing'
- * Security testing - testing how well the system protects against unauthorized internal or external access, willful damage, etc; may require sophisticated testing techniques.
- * Compatibility testing - testing how well software performs in a particular hardware/software/operating system/network/etc. environment.
- * Exploratory testing - often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it.
- * Ad-hoc testing - similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.
- * Context-driven testing - testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life-critical medical equipment software would be completely different than that for a low-cost computer game.
- * User acceptance testing - determining if software is satisfactory to an end-user or customer.
- * Comparison testing - comparing software weaknesses and strengths to competing products.
- * Alpha testing - testing of an application when development is nearing completion; minor design changes may still be made as a result of such testing. Typically done by end-users or others, not by programmers or testers.
- * Beta testing - testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end-users or others, not by programmers or testers.
- * Mutation testing - a method for determining if a set of test data or test cases is useful, by deliberately introducing various code changes ('bugs') and retesting with the original test data/cases to determine if the 'bugs' are detected. Proper implementation requires large computational resources.

11. What are 5 common problems in the software development process?

- * Solid requirements - clear, complete, detailed, cohesive, attainable, testable requirements that are agreed to by all players. Use prototypes to help nail down requirements. In 'agile'-type environments, continuous coordination with customers/end-users is necessary.
- * Realistic schedules - allow adequate time for planning, design, testing, bug fixing, re-testing, changes, and documentation; personnel should be able to complete the project without burning out.
- * Adequate testing - start testing early on, re-test after fixes or changes, plan for adequate time for testing and bug-fixing. 'Early' testing ideally includes unit testing by developers and built-in testing and diagnostic capabilities.
- * Stick to initial requirements as much as possible - be prepared to defend against excessive changes and additions once development has begun, and be prepared to explain consequences. If changes are necessary, they should be adequately reflected in related schedule changes. If possible, work closely with customers/end-users to manage expectations. This will provide them a higher comfort level with their requirements decisions and minimize excessive changes later on.
- * Communication - require walkthroughs and inspections when appropriate; make extensive use of group communication tools - e-mail, groupware, networked bug-tracking tools and change management tools, intranet capabilities, etc.; insure that information/documentation is available and up-to-date - preferably electronic, not paper; promote teamwork and cooperation; use prototypes if possible to clarify customers' expectations.

12. What is software 'quality'?

* Quality software is reasonably bug-free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable. However, quality is obviously a subjective term. It will depend on who the 'customer' is and their overall influence in the scheme of things.

* Management/accountants/testers/salespeople, future software maintenance engineers, stockholders, magazine columnists, etc. Each type of 'customer' will have their own slant on 'quality' - the accounting department might define quality in terms of profits while an end-user might define quality as user-friendly and bug-free.

13. What is 'good code'?

* 'Good code' is code that works, is bug free, and is readable and maintainable. Some organizations have coding 'standards' that all developers are supposed to adhere to, but everyone has different ideas about what's best, or what is too many or too few rules. There are also various theories and metrics, such as McCabe Complexity metrics. It should be kept in mind that excessive use of standards and rules can stifle productivity and creativity. 'Peer reviews', 'buddy checks' code analysis tools, etc. can be used to check for problems and enforce standards. For C and C++ coding, here are some typical ideas to consider in setting rules/standards; these may or may not apply to a particular situation:

* Minimize or eliminate use of global variables.

* Use descriptive function and method names - use both upper and lower case, avoid abbreviations, use as many characters as necessary to be adequately descriptive (use of more than 20 characters is not out of line); be consistent in naming conventions.

* Use descriptive variable names - use both upper and lower case, avoid abbreviations, use as many characters as necessary to be adequately descriptive (use of more than 20 characters is not out of line); be consistent in naming conventions.

- * Function and method sizes should be minimized; less than 100 lines of code is good, less than 50 lines is preferable.
- * Function descriptions should be clearly spelled out in comments preceding a function's code.
- * Organize code for readability.
- * Use whitespace generously - vertically and horizontally.
- * Each line of code should contain 70 characters max.
- * One code statement per line.
- * Coding style should be consistent through a program (eg, use of brackets, indentations, naming conventions, etc.)
- * In adding comments, err on the side of too many rather than too few comments; a common rule of thumb is that there should be at least as many lines of comments (including header blocks) as lines of code.
- * No matter how small, an application should include documentation of the overall program function and flow (even a few paragraphs is better than nothing); or if possible a separate flow chart and detailed program documentation.
- * Make extensive use of error handling procedures and status and error logging.
- * For C++, to minimize complexity and increase maintainability, avoid too many levels of inheritance in class hierarchies (relative to the size and complexity of the application). Minimize use of multiple inheritance, and minimize use of operator overloading (note that the Java programming language eliminates multiple inheritance and operator overloading.)
- * For C++, keep class methods small, less than 50 lines of code per method is preferable.
- * For C++, make liberal use of exception handlers.

14. What is 'good design'?

* 'Design' could refer to many things, but often refers to 'functional design' or 'internal design'. Good internal design is indicated by software code whose overall structure is clear, understandable, easily modifiable, and maintainable; is robust with sufficient error-handling and status logging capability; and works correctly when implemented. Good functional design is indicated by an application whose functionality can be traced back to customer and end-user requirements. For programs that have a user interface, it's often a good idea to assume that the end user will have little computer knowledge and may not read a user manual or even the on-line help; some common rules-of-thumb include:

- * The program should act in a way that least surprises the user
- * It should always be evident to the user what can be done next and how to exit
- * The program shouldn't let the users do something stupid without warning them.

15. What is SEI? CMM? CMMI? ISO? IEEE? ANSI? Will it help?

* SEI = 'Software Engineering Institute' at Carnegie-Mellon University; initiated by the U.S. Defense Department to help improve software development processes.

* CMM = 'Capability Maturity Model', now called the CMMI ('Capability Maturity Model Integration'), developed by the SEI. It's a model of 5 levels of process 'maturity' that determine effectiveness in delivering quality software. It is geared to large organizations such as large U.S. Defense Department contractors. However, many of the QA processes involved are appropriate to any organization, and if reasonably applied can be helpful. Organizations can receive CMMI ratings by undergoing assessments by qualified auditors.

* Level 1 - characterized by chaos, periodic panics, and heroic efforts required by individuals to successfully complete projects. Few if any processes in place; successes may not be repeatable.

* Level 2 - software project tracking, requirements management, realistic planning, and configuration management processes are in place; successful practices can be repeated.

* Level 3 - standard software development and maintenance processes are integrated throughout an organization; a Software Engineering Process Group is in place to oversee software processes, and training programs are used to ensure understanding and compliance.

* Level 4 - metrics are used to track productivity, processes, and products. Project performance is predictable, and quality is consistently high.

* Level 5 - the focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required.

at Level 1, the most problematical key process area was in Software Quality Assurance.

* ISO = 'International Organisation for Standardization' - The ISO 9001:2000 standard (which replaces the previous standard of 1994) concerns quality systems that are assessed by outside auditors, and it applies to many kinds of production and manufacturing organizations, not just software. It covers documentation, design, development, production, testing, installation, servicing, and other processes. The full set of standards consists of: (a)Q9001-2000 - Quality Management Systems: Requirements; (b)Q9000-2000 - Quality Management Systems: Fundamentals and Vocabulary; (c)Q9004-2000 - Quality Management Systems: Guidelines for Performance Improvements. To be ISO 9001 certified, a third-party auditor assesses an organization, and certification is typically good for about 3 years, after which a complete reassessment is required. Note that ISO certification does not necessarily indicate quality products - it indicates only that documented processes are followed. Also see <http://www.iso.ch/> for the latest information. In the U.S. the standards can be purchased via the ASQ web site at <http://e-standards.asq.org/>

* IEEE = 'Institute of Electrical and Electronics Engineers' - among other things, creates standards such as 'IEEE Standard for Software Test Documentation' (IEEE/ANSI Standard 829), 'IEEE Standard of Software Unit Testing (IEEE/ANSI Standard 1008), 'IEEE Standard for Software Quality Assurance Plans' (IEEE/ANSI Standard 730), and others.

* ANSI = 'American National Standards Institute', the primary industrial standards body in the U.S.; publishes some software-related standards in conjunction with the IEEE and ASQ (American Society for Quality).

* Other software development/IT management process assessment methods besides CMMI and ISO 9000 include SPICE, Trillium, TickIT, Bootstrap, ITIL, MOF, and CobiT.

16.What is the 'software life cycle'?

* The life cycle begins when an application is first conceived and ends when it is no longer in use. It includes aspects such as initial concept, requirements analysis, functional design, internal design, documentation planning, test planning, coding, document preparation, integration, testing, maintenance, updates, retesting, phase-out, and other aspects.

17.Will automated testing tools make testing easier?

* Possibly For small projects, the time needed to learn and implement them may not be worth it. For larger projects, or on-going long-term projects they can be valuable.

* A common type of automated tool is the 'record/playback' type. For example, a tester could click through all combinations of menu choices, dialog box choices, buttons, etc. in an application GUI and have them 'recorded' and the results logged by a tool. The 'recording' is typically in the form of text based on a scripting language that is interpretable by the testing tool. If new buttons are added, or some underlying code in the application is changed, etc. the application might then be retested by just 'playing back' the 'recorded' actions, and comparing the logging results to check effects of the changes. The problem with such tools is that if there are continual changes to the system being tested, the 'recordings' may have to be changed so much that it becomes very time-consuming to continuously update the scripts. Additionally, interpretation and analysis of results (screens, data, logs, etc.) can be a difficult task. Note that there are record/playback tools for text-based interfaces also, and for all types of platforms.

* Another common type of approach for automation of functional testing is 'data-driven' or 'keyword-driven' automated testing, in which the test drivers are separated from the data and/or actions utilized in testing (an 'action' would be something like 'enter a value in a text box').

Test drivers can be in the form of automated test tools or custom-written testing software. The data and actions can be more easily maintained - such as via a spreadsheet - since they are separate from the test drivers. The test drivers 'read' the data/action information to perform specified tests. This approach can enable more efficient control, development, documentation, and maintenance of automated tests/test cases.

* Other automated tools can include:

* Code analyzers - monitor code complexity, adherence to standards, etc.

* Coverage analyzers - these tools check which parts of the code have been exercised by a test, and may be oriented to code statement coverage, condition coverage, path coverage, etc.

* Memory analyzers - such as bounds-checkers and leak detectors.

* Load/performance test tools - for testing client/server and web applications under various load levels.

* Web test tools - to check that links are valid, HTML code usage is correct, client-side and server-side programs work, a web site's interactions are secure.

- Other tools - for test case management, documentation management, bug reporting, and configuration management.

For more Software Testing Resources, please visit

<http://www.OneStopTesting.com>

Join largest Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Software Testing Interview Questions at

<http://www.CoolInterview.com>

and

<http://www.TestingInterviewQuestions.com>

and

<http://www.NewInterviewQuestions.com>

World's Largest Portal on Software Testing Information & Jobs -

<http://www.OneStopTesting.com>

Join Software Testing Community at

<http://groups.yahoo.com/group/OneStopTesting/>

Over 5,000 Testing Interview Questions at

<http://www.CoolInterview.com>