Software Testing and Quality Assurance

Assignment -1 **- Kondakindi Pranitha Reddy**

2.1 & 2.2) **QUICK TEST PROFESSIONAL**

- Quick Test Professional is a Mercury Interactive product taken over by HP.
- It's a Functional Testing tool that is used to check the functionality of the application. It is also called as Regression Testing tool.
- It supports all the technologies like c,c++,java,dotnet,vb,php etc
- It can record all the manual test cases in VB Script.

Scripting Language: The script is inVisual basic and it works only on windows platform.Vb script is not case sensitive.It is used for implementing the programming logic.

QTP consists of following contents:

1)Test Pane: It contains Keyword view and Expert view tabs

- Keyword View: It represents the operation performed through keyboard or mouse in English language
- Expert View: It represents the operation performed in Vb Script

2)Active Screen: It provides snapshot of your application as it appeared when you perform a certain step during the recording session.

3)Data Table Pane: It will assist the user in parameterizing user tests. We can maintain test data in data table and pass on this data to the script for multiple iterations.

4)Debug Viewer Pane: It will assist you in debugging your test.

QTP Life Cycle

- Prepare to Record
- Record operation on application
- Enhancing test
- Debugging test
- Running test

- Analyzing the test results
- Reporting Defects

The Testers convert all manual functional test case steps into automation scripts in two ways

1) Recording/Generating/Descriptive Programming

2)Inserting CheckPoints

Recording: Testers record all manual operations on the application by recording. In qtp they are 3 recoding modes

- **Normal Mode:** In this mode, qtp records mouse and keyboard operations with respect to object or window. By default qtp is always in normal mode.
- **Analog Mode:** In this mode qtp records mouse pointer movements relative to the screen or relative to specific window. This is optional recording mode. Testers generally use this mode wherever they require drawing, painting and signatures. We can use shortcut key as Alt+Shift+F3
- **Low Level Recording:** In this mode, qtp records advanced test objects or unrecognized objects. We can use short cut key as Ctrl+Shift+F3

Step Generator: Testers not only record scripts but also generate scripts by using step generator

Navigation to open step generator: Click F7 or insert>step generator

- **Functions:** These are subprograms which must return a value to the calling place.

  a) Built-in functions/ System functions

  b) User defined functions

- **Test Objects:** These are run time objects while we are performing operations.
- **Utility Objects:** These objects are reserved objects in quick test professional and are used to report preferences during run time.

  a) Crypt object: It is used to encrypt a string which can be protected. It used encrypt method. For example it is used to hide the password for security reasons.

  b) Reporter: It is used to get the result in result window.

c)Systemutil: It is used to invoke executable application

Synchronization Point: Sometimes our application operations are taking some valid delay to return output. During this type of operations, testers are inserting synchronization point to define time mapping between build and tool.

a) wait(): This function defines a fixed waiting time.

b) Synchronization point: We can use this option to synchronize the tool and build depending on properties of objects.

c) Global Settings: The testers are enhancing the synchronization timeout in settings of qtp as per the requirement.

Inserting Checkpoints: Check points are used to compare testers expected value with the build actual value. These check points are also called as Verification points. Qtp supports following check points.

- **Standard Checkpoint:** It is used to check property value of an object in our application. Such as radio buttons, combo boxes etc

  Standard checkpoints on web applications

  a) Page Checkpoint: This is used to check load time, number of links, number of images on web applications

  b) Image Checkpoint: It is used to check properties of web application image.

  c) Table Checkpoint: It is used to check the content of tables of web application.

- **Bitmap Checkpoint:** It is used to compare expected area image with actual area image.

- **Text Checkpoint:** It checks that a text string is displayed in the appropriate place in the application. It has four sub check points

  a) Match case: It checks case sensitive of the string

  b) Exact match: It does not check case sensitiveness

  c)Text not displayed: It is used for negative testing

  d)Ignore space: It ignores spaces in between words

- **TextArea Checkpoint:** It does not support web applications. It checks that a text string is displayed within a defined area in a window application.
- **Database Checkpoint:** It is used to check the contents of database accessed by our application.
- **Accessibility Checkpoint:** It identifies areas of website hidden files. This checkpoint checks whether the website is developed according to www standards or not.
- **Xml Checkpoint:** It is used to check the data content in xml documents or files.

Data Driven Testing/Parameterization:

The re-execution of single script with multiple test data is called as data driven testing or parameterization. They are two ways:

1) Dynamic entry of test data: In this method we are using for loop to iterate the operations for specified number of times sequentially.

2) Maintain data in external files: Testers maintain data into data tables without hardcode into the script.

a)Maintain data into the data table: Testers are maintain data into the data table of qtp with the help of data driver, they pass on the test data to the script. So that qtp will execute the operation on the application.

b) Import/export data from external sheets: Testers are importing/exporting test data from external sheets into the data tables of qtp with the help of data drivers, they pass on the test data to the script. So that qtp will execute the operation on the application.

Recovery Scenario Manager(Exception Handling)

It is a problematic situation faced by the qtp during the execution where it needs to recover itself. So that testers give some scenarios to reliable sources. Then the qtp will use that solution and recover form that situation and continues its execution. Extension to this files are ".qrs"(quick text recovery scenario).

Actions

Action is a set of script statements to perform a specific task.

Types of actions

a) Normal Action: We can utilize once the each action.

b) Re-usable Action: We can utilize each action number of times.

c) External Action: We call re-usable actions into other tests called as external actions.

Split Actions

The testers are able to split test into individual actions. They are two types of split actions

a) Independent of each other: These actions are independent of each other. We need to call each action individually.

b) Nested Actions: These actions are dependent on each other. We need not to call each action. Once we call parent action automatically child action will come along with that.

Virtual Object Manager

It is a feature provided by qtp used for managing the virtual concept.

Object Repository

It is a storage place where qtp store the objects information and it will also acts as an interface between the test script and the application under test in order to identify the objects during execution. Shortcut key Ctrl+R

Types of Object Repository

- Per-action/Local Repository

  By default qtp is in per test mode. They qtp will create and manage an individual repository separately for each and every action. Extension is .mtr(module test repository)

- Shared Object

  Qtp stores the object information in object repository and that can be associated with multiple tests. The extenstion is .tsr(test script repository)

Object Identification

Sometimes our application screens are maintaining more than one similar objects while running test, testers are enhancing physical description of that objects in the object identification option of qtp. It is based on two types of properties and an ordinal identifier.

Types of properties:

a)Mandatory Properties

b)Assistive properties

2.3) Strengths of qtp tool

- It is easy even for novice users to understand QTP and start adding test cases.
- QTP uses Vbscript which is English-like and very easy to learn and gain expertise
- It is easy to maintain data driven through configurations.
- It supports various addins like Java, Oracle, Web services, SAP, .NET, Siebel, PeopleSoft etc.
- Support for record and playback and ability to edit scripts after recording.

Weekness of qtp tool

- Cost is high – License and maintenance.
- Slow execution in comparison with other tools like Selenium.
- You get support from HP only when you renew the license. You need to buy different addins – Java, Oracle, SAP, .Net, Seibel, Peoplesoft etc.

2.4) References & Links:

www.onestoptesting.com/qtp/

http://www.softwaretestinghelp.com/

http://en.wikipedia.org/wiki/HP_QuickTest_Professional

3) Article Report: What is Software Testing? And Why Is It So Hard?

3.1) Software Engineering is "the process of executing a software system to determine whether it matches its specifications and executes in its intended manner."

Though an adequate amount of time is spent on software testing, recognizing its importance, some bug may not be fixed because of various reasons like some part of code may not be tested, some combination of inputs were untested, user's operating environment interface was untested, excetra.

Testing can be divided into four phases:

i) Modeling the software's environment.

ii) Selecting test scenarios.

iii) Running and evaluating test scenarios.

iv)Measuring testing progress.

i) Modeling the software's environment:

In this phase, testers identify and simulate the environment that a software system uses and enumerate the inputs that can cross the interface. Four common interfaces are,

a) Human Interfaces: These includes all common methods that people interact with the system.

b) Software Interfaces: These includes how software uses operating system, run-time library, database. Error messages are also to be tested.

c) File System Interfaces: These interfaces exists whenever software reads or writes data to external files. Thus testers must generate files containing both legal and illegal, variety of texting and formatting.

d) Communication Interfaces: These interfaces allows direct access to physical devices and they require a communication protocol. Testers must be able to produce both legal and illegal protocol streams, different combination of commands and data in a proper packet format.

Testers must also understand the user control that falls outside the software.

For selecting single value variables, testers often use boundary value partitioning technique, in which variables are tested around their boundaries. For choosing values for multiple variables, tester must consider entire cross product of values. To overcome a sequence generation problem, testers define model of a language. The model is a representation that describes how input and event symbols are combined to make syntactically valid words and sentences.

ii) Selecting Test Scenarios:

Testers are interested in execution paths rather than individual statements of the code. Execution paths are the sequence of code statements that represent the execution of software. Testers use best possible test data adequacy criteria. Testers will seek the test sets that will find most of the bugs. Test data adequacy criteria concentrates on either execution path test criteria or input domain criteria but rarely both.

In execution test path criteria, testers mostly test the paths that cover control structure. The dataflow family of test data adequacy criteria test the data that is moved from one location to another. In fault seeding method, faults are intentionally inserted into code. While finding the seeded errors,testers will also find real errors.

Simple coverage of an interface to more complex statistical measurement forms the coverage for input data coverage.

iii) Running and Evaluating Test Scenarios:

Testers try to automate the test scenarios as much as possible because manual application of test scenarios is labor intensive and cost oriented.

In evaluation process, the output generated by the software for the given test scenario is compared with that of the output as documented by specification. Testers pursue formalism and embedded test code methods for evaluation process.

Formalism primarily concentrates on formalizing the way specifications are written and the way design and code are derived from them. Expressing the specifications in formal way makes the comparison easier. Informal expression of a good specification is still helpful because without a specification testers may report unspecified features as bugs.

Embedded test code consists of two types. One, exposes certain internal objects or states to make external oracle rectify errors easily. The other complex type of embedded test code is self testing programs. Testers code multiple solutions and one solution check the other solutions.

Once bugs are reported to developers and developers generate a new code. Regression testing (retesting) will consume more number of resources. Testers work close with the developers to prioritize and minimize regression testing.

iv) Measuring Testing Progress:

Testers use the structural and functional testing completeness to measure testing progress. Testers want the quantitative measures for how many bugs are left, what is the probability of arrival of the bug, so that they can stop testing.

Testability determines the testing complexity. If a product has high testability, it is easy to test. On the other hand, if it has less testability indicates more number of tests for a conclusion.

Reliability models are the mathematical models of test scenarios and failure data that attempt to predict future failure patterns based on the past data.

3.2) I have learned how to measure the testing progress. Structural and functional testing can be used to measure the testing progress.

In functional testing, the test selection method and test scenarios must be based on attributes of the specification or operational environment but not on the attributes of code. It can also be called as black box testing.

In structural testing, inputs must be based on code or data structures. It is also called as white box testing.

3.3) In Measuring Testing progress, author says that testers can stop testing if they have quantitative measure of bugs and their probability of occurrence. In my view, the number of bugs unaddressed must be zero, because even one unaddressed bug in the software can collapse the entire system and result in a huge loss.

3.4) I frequently use execution test path criteria to test my code. I usually examine the code and its branching structures and find the input values(if any) for which code fails to handle them.

3.5) Can we find a measure which tells us how many bugs are left in the code? On what basis will the testability be given to a code?