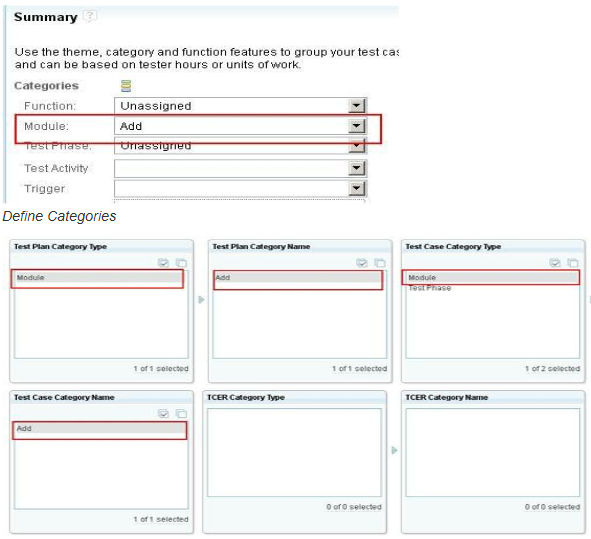
### Best practices

### Using test artifacts categories for organizing and generating effective reports:



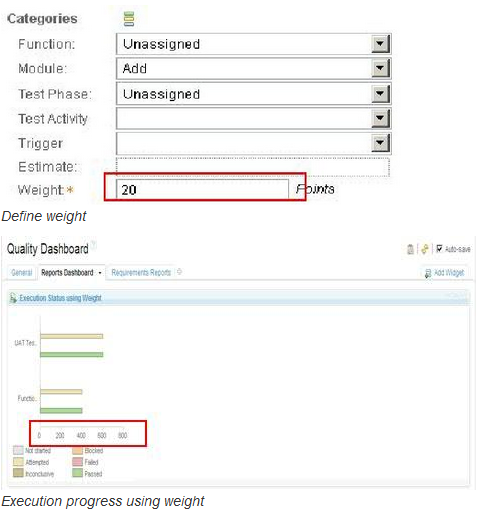
This is one of the very good features of the RQM where you can define the category for each of the test artifacts. Category is basically in an attribute of the test artifacts. When I am saying the test artifact means like test cases, test suits or test plan or test scrapes or execution record all are in RQM it is a test artifacts. So how it will be helpful, it will physically help to group the test cases test plan.   
  
Example1

If we have test cases which is part of the QAT so you can define the phases the category and that phase have a value called unit exceptions. So whenever you define the test cases you can associate the test cases with the QAT test. As well as you can define like n number of categories for modules for a product or release.

Example2

Like here there is an option where you can define the multi valued category also. For example say if your test case is part of system testing phase as well as part of QAT phase now you want to see your report like which are the common test cases which is far presentation to QAT so you can define that category see which is a multi-valued, where you can have a two option to select. Currently U junction in combo box like which is highlighted only one option but if the category is defined as the multi value and you can have an option with the chat box. So you can select both the options, so that way you can define this category as well as the category also help you to define the reporting part also, it will help you to fettering test artifact, it will help to group the artifacts, as well you can define the soft category also for each of the best test artifacts. Typically you can define the financial relationship and as well as you can have if you want to filter this category when telecasting option is available in the artifact.

1. **Defining Weight**

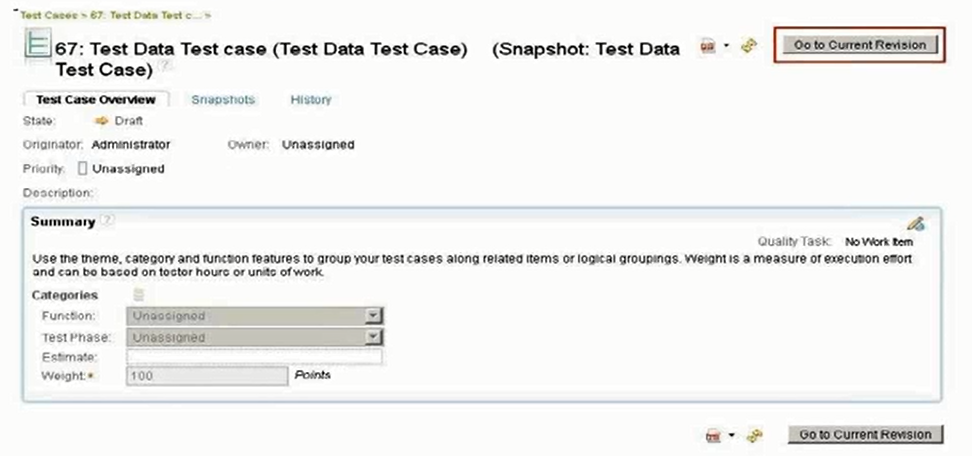


There is a one more feature called weight, weight is basically define the complexity of the test case where you can define the typically the numeric number on this. How it will help, so for example, client says that if he says we have a 500 test cases and client want to understand like what is the execution time, how much time it will take to execute to find the test cases. So in that case actually this is a real scenario which was in our project it was asked that execution of this 500 test cases of this how much time it will take.

So what we have done is we define the weight attribute uses to the test cases based on the complexity. If it is a simple then this is in a weight, it is a complex time my weight is a 500 time or if it is very complex then my weight is 1000. So based on that actually we define for test cases weight is 100 then it will take 2mins and if the weight is 500 then it will take 4mins, if it is 1000 in that case this much time to execute test cases. So based on the category again we have define the grouping of all the test cases, we have calculated the weight that how many weight for the 100, and how many for the 500, and how many for the 1000 and based on this we retrieve some of the estimations that this is how a total to execute 500 test cases this much time it will take. Basically it is not addition of weight that is basically defining how you can define the complexity of the test cases and based on that you can define the timings. One more like in the real scenario what we used to do is like to define the test suit, for each test suit we have a test cases and for each test cases we have a test scripts.   
  
So say if you define that weight at suit level also, if we have two test cases and two test cases is a part of one test suit and when you run that test suit in that case either we have to run the both the test cases or only we can run the one test cases and for some reasons we are not able to run the another test cases then it will give you what is the execution progress and how much that test suit has run. So here that will also help you to understand that what is the execution progress of your test cases is. This is also only defining using the weight attribute, as well as it will help you into for the report also. If you want to know the report that what is your execution progress using the weight, in that case if you see here on the X sector which you say the total weight is and Y is basically category. So here like for functional section there are weight 400 that already passed and there are weight 400 that is still it is blocked or I mean it is stayed in the bottom. So give me a progress of all the test cases using the weight also.

1. **Snapshots**

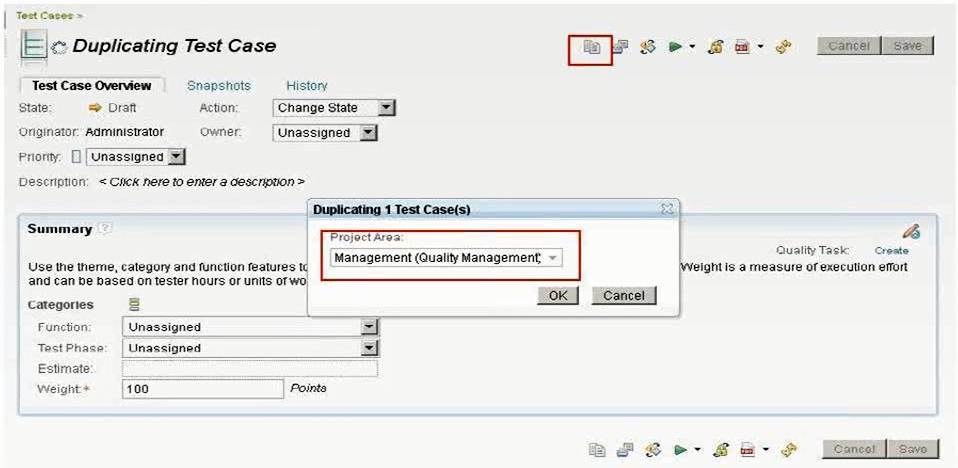
Snapshots are non-editable and cannot be replacing the current baselines instead it serves as reference of previous version.



This is one more feature called snapshot which is basically is related to creating the version but it is not really true version management. If we are see working on any test plan or test cases and if we want to create a copy of the hidden view to maintain the backup of the old one in that case we can create a snapshot. But snapshot is basically a non-editable, we cannot modify anything in the snapshot copy of the test artifact but in case if we want to refer the older copy of the test artifact then you can go to the snapshot copy and you can have a look at it.   
  
Whenever you create a snapshot basically the copy all the data of that test cases and test plan are in the new test artifact. You can create a snapshot either in the same project area or in the different project area. In the same project area if you create it would basically it will give the reference of the previous test artifact. For example, if we have test cases and the test cases associated with the test scripts and if you create a snapshot of the test cases then it won’t create a new test scripts it will refer the test scripts which already exists for that particular test case.   
  
If you create a snapshot in the different project area then it will create the other attribute also. So this is something which we use like many times it will happen like we want to maintain the old copy for the test plan or test cases and instead of relating the or instead of over writing of the same copy, we used to maintain the version by using a snapshot and we create a new one. But only thing is we cannot after creating the new one if we want to move to the old version then it cannot be done so this is based on our requirement like it was very helpful we have create number of snapshots for our project so we thought just give the idea which can be used for the maintaining the version.

1. **Copy Utility**

We can copy the artifacts into same or different project areas to keep baselines of test artifacts to replace current with baseline user need to do delinking and linking manually.



Create the copy of the test artifacts then there is a very simple option where you can copy the basically even career duplicate of the test cases and test plan so again in a same project area or in a different project area. So again if the test cases proceeds have a link with the test scripts and when you copy the test cases then it won’t create a new test cases but it create a reference with that test scripts.   
  
So this is also a very powerful feature like if we want to say copy of 100 of test case or say if you want to copy a duplicate 100 of test cases then you can select the option in the RQM like who are browsing of all the 100 test cases, select all the 100 test cases and just click on the option where all the selected test cases it’s just in single click it will create all the copy of 100 test cases.

1. **One Test Script per Test Case**:

RQM provides flexibility of attaching many test scripts to test case for execution. However at run time we can select any one of the test script to run.

It’s strongly recommended to use one test script per test case. We can use more than one test script per test case only in following scenario

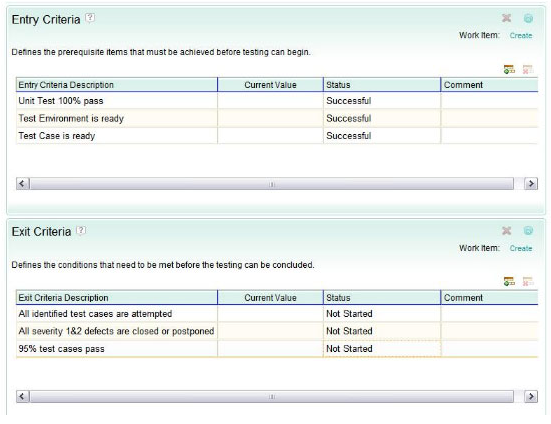
Scenario: Use more than one test script if you are running your test case in different platform like Linux, Windows etc. Ex: your automation script may vary depending upon operating system and underlining hardware and software. In this scenario you choose test script at execution time which you planned to run in respective platform.

1. **Entry criteria and Exit criteria**

Test Plan is basically talk about planning of your test effort and execution. The test plan can be configured to meet the needs of your team.

Test Plan also provides two sections called Entry criteria and Exit criteria. These two criteria help to deliver quality product and plan your deliverables accordingly.

Before you start your test plan execution you can evaluate your entry criteria whether it is successful or unsuccessful and decide upon whether you can go for further testing or not. Similarly you can also evaluate after your test execution is completed using exit criteria.



1. **Using Keywords, Test Data and Execution Variables**

**Keywords** are nothing but reference point to execution step and can be reused these keywords in several scripts there it avoids re-writing the same script/steps in script.

**Test Data** is simple set of records written in spread sheet and can be used data in our manual test script.

Ex: The same script is run for different set of test data, sometimes called as data driven testing.

**Execution Variable** nothing but variables, these values keeps changing from one script to another. This variable can be passed across different scripts.

Ex: In login script. The login name can be stored in variable and can be used in another script for finding out login details.

