

QA Paramedic - Emergency Requirements Traceability

By Randy Raymond

Scenario

You are a consultant, contractor, test lead or test manager and have been brought in to save a critically injured testing project. The project has crashed and burned at least once, maybe several times, and the last crash was the last straw. Senior management is desperate to find a way to succeed. Now it's your job to save the day. This is going to be tough.

On Scene

You arrive on scene and start taking inventory of your surroundings while observing everything around you. Your patient, the project, is on fire and burning in several places. The flames are not too high and each fire has someone spraying the flames with a fire extinguisher. "Good, one body on each hot spot. That will hold us for the moment" you say to yourself.

The previous test manager is pretty banged up sitting with her head back against the wall. She is staring into space with her face expressionless, drained of color, and no visible emotion. She's mumbling something about being run over by a virtual steamroller.

You see several testers covered in rumpled clothing slumped in their chairs staring off in the distance. They're exhausted. No amount of caffeine from coffee or energy drinks will revive them. Their cholesterol levels are out of control from eating pizza three meals a day seven days a week for the last 2 months. Not quite walking dead.

One tester is just sitting on the ground staring blank-faced into oblivion. His eyes are not blinking but he does seem to be breathing.

You have been here before. That's why you are here now. At first look, this won't be too difficult after all. You send everyone home for the weekend telling them to get plenty of rest and to not come back until Monday. Out of habit some of the team resists but you insist they go home for the weekend.

Triage

To triage is to prioritize on the basis of severity in order to determine who should be treated first. Sounds like functional risk-based test prioritization doesn't it?

Your triage identified that the process is the problem not the people although the people look pretty bad because of the broken process.

Your patient in this case is the process and in the course of treating the process you will also be treating the people who work in the process. When you arrived on scene you saw the usual signs of process trauma and know exactly how to treat the injury. Your training and experience will revive the patient and make the process productive again.

Monday dawns and the team are starting to arrive at the office. You speak with the test manager and arrange to interview her and as many of the team as you can for the next few days. You are conducting due diligence as part of your assessment procedures.

Virtual Steamroller

The interview with the test manager is pretty comprehensive. She has tried to do all of the right things but many influential forces from executives, development, project management, business users and other stakeholders ran her over in favor of hitting the delivery date. The other stakeholders have told her we're doing everything right in the process but we have to cut corners to meet the delivery date.

Your interviews with the test manager and several members of the test team shows they have very comprehensive test cases. In fact you find out that there are also well-documented and comprehensive requirements. There are ten requirements documents and there are 10 test case documents.

You ask to see the requirements trace matrix (RTM). You get the answer "We don't have one. That was a corner the very influential and powerful PM told us to cut in order to meet the delivery date. With a 1-1 ratio of requirements documents to test case documents it is implied that test coverage has been achieved so there is no need to do an actual trace between the sets of documents. The PM declared we had performed traceability when we created the test case documents."

The test manager was run over by the virtual steamroller along with her entire team.

Read the Instructions on the Fire Extinguisher

All fire extinguishers have written and pictorial instructions on the outside. Point the fire extinguisher at the base of the fire. If you can't read, the picture of pointing the fire extinguisher at the base of the fire is worth a thousand words in any language.

When you arrived on scene you noted that there was a tester spraying a fire extinguisher at the flames of each hot spot. That's what the powerful and influential PM told them to do so that's what they did.

The problem is, the fire extinguisher should be pointed at the base of the fire. You've identified the problem and now the problem can be solved.

In our case, testing processes and best practices is the equivalent of spraying the base of the fire. Best practices says you need to have a RTM. You start directing the testers to create an RTM (spray the base of the fire). You and the executive that sent you in to fix the testing problems will deal with the PM later.

Solution

We have to get back to work quickly but we can't stop testing to fix the process. The solution will be to build an RTM that also serves as a test management and test reporting tool. The activity of creating traceability between tests and requirements will build a vast wealth of application knowledge in the testing team while creating a resource that accelerates the learning curve of future testers. We will in effect be taking tribal knowledge and creating corporate memory.

We will need to provision the requirements documents and test case documents to facilitate traceability then feed information to the RTM. VBA macros will do the work after the test team has finished their traceability task in the documents.

RTM

The RTM will be built in a spreadsheet and have the following tabs and columns:

Tab: Master Summary

Column	Description
Requirement name	A text human readable name for the requirement
Requirement ID	A unique sequential serial number for the requirement
Requirement priority	High, medium, or low. This value is assigned by the business stakeholder
Test case ID	A unique sequential serial number for the test case
Execute	Indicates that the test case will be executed in the current test cycle. Typical value would be Yes, No, "x" or some other indicator to show the test case needs to be executed.
Pass / Fail	Indicates the test case has been tested and has passed or failed. An empty cell indicates the test case is not tested.
Defect ID	If a test case fails this column will contain the defect number assigned by the defect tracking tool
Defect Severity	Indicate the severity assigned to the defect in your defect tracking tool

Table 1. RTM Master Summary Columns

Tab: List of RQT Docs

Column	Description
Path	Path file location where you have stored the requirement document
Document Name	Name of the document to be included in the trace matrix

Table 2. RTM Master Summary Columns

Your spreadsheet should look as shown:

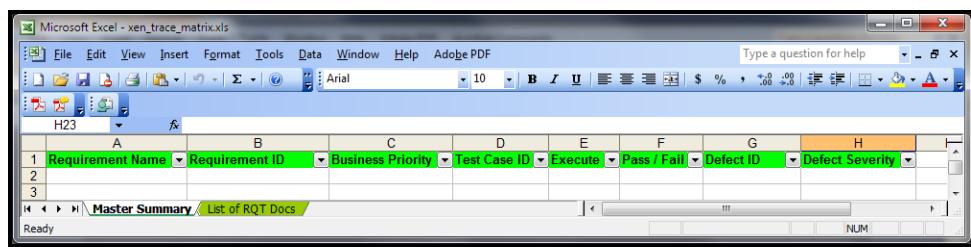
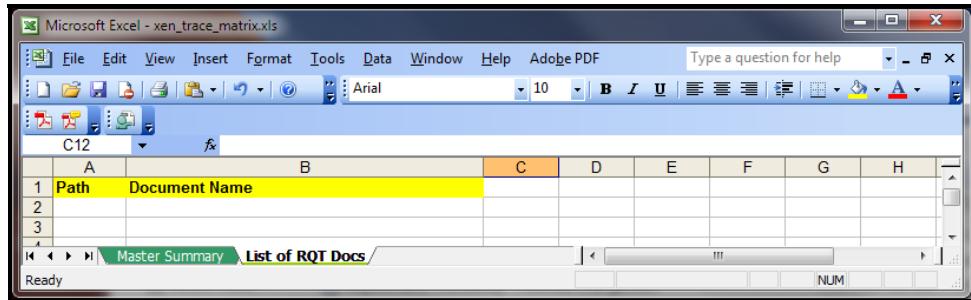



Figure 1. RTM Spreadsheet

Note that the AutoFilter feature is enabled. The AutoFilter feature is a cornerstone for reporting, test planning, and impact analysis. Search "AutoFilter" in Excel help if you need more information on this feature.

Provision Documents

We will leverage three features of MS Word to link test cases to requirements - tables, bookmarks, and hyperlinks. There will be a lot of elbow grease work but the results will be well worth the effort.

Here is how we will connect the two documents:

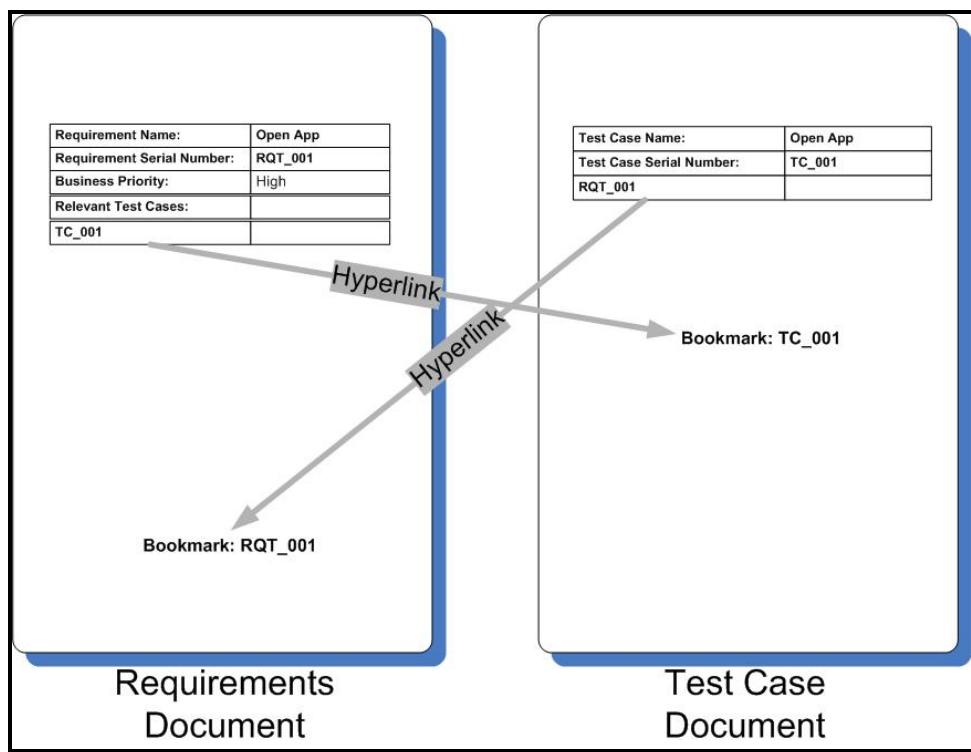


Figure 2. Connecting the documents

Original Requirements Document

The requirements document is the usual MS Word document to describe the features in pictures or text. It will look something like this oversimplified example:

Open App

Phasellus malesuada velit sed tortor ultrices sit amet convallis enim commodo. Curabitur velit uma, auctor vitae bibendum a, adipiscing sit amet dui. Duis sit amet enim eu erat auctor rutrum. In hac habitasse platea dictumst. Suspendisse sed rutrum massa. Cras malesuada mattis nunc, a blandit augue blandit id. Fusce nec dolor id metus dictum commodo. Aliquam volutpat nisi vel nibh adipiscing aliquam. Donec imperdiet metus vel enim faucibus fermentum. Sed a diam nisi. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin dignissim libero nec enim malesuada eu viverra quam vestibulum. In pulvinar, tortor at varius viverra, justo turpis pharetra metus, eu accumsan lacus leo a ante. Proin rutrum ipsum pharetra elit egestas porta. Vivamus suscipit vehicula dapibus.

Figure 3. Requirement document sample

Original Test Case Document

The test case document is the usual MS Word document to describe the features in pictures or text. It will look something like this oversimplified example:

Figure 4. Test case document sample

Hard Work of Creating Traceability

This next part is where all of the hard work is performed. Now you have to link test cases to requirements. An efficient method to link test cases to requirements is to use tables, bookmarks, and hyperlinks.

Performing the hard work of reading all of your requirement and test documents will create a great deal of expertise in the application. You and your team will need this knowledge as you link test cases to requirements.

Step 1 - Provision your documents

Review each requirement document and each test case document by inserting a table every place a requirement is called out and all places where a test case is called out.

Provisioned Requirements Document

Add a table to each place in the requirements document that represents features and functionality that need to be tested and verified.

Open App	
Requirement Name:	
Requirement Serial Number:	
Business Priority:	
Relevant Test Cases:	
Phasellus malesuada velit sed tortor ultrices sit amet convallis enim commodo. Curabitur velit urna, auctor vitae bibendum a, adipiscing sit amet dui. Duis sit amet enim eu erat auctor rutrum. In hac habitasse platea dictumst. Suspendisse	

Figure 5. Requirement document with table

Provisioned Test Case Document

Add a table each place in the test case document that represents a test of some application feature or requirement.

Test Open App	
Test Case Name:	
Test Case Serial Number:	
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc et dapibus velit. Fusce eu quam non est convallis faucibus eget eu eros. Pellentesque nec sem ut risus cursus	

Figure 6. Test case document with table

Step 2 - Add Names and Serial Numbers

The name should be something human readable which could or could not be identical to the paragraph title in the document. Serial numbers are unique numbers you will use to identify each requirement or test case. Bookmarks described later will help you to enforce this unique number.

Named and Numbered Requirement

Open App	
Requirement Name:	Open App
Requirement Serial Number:	RQT_001
Business Priority:	
Relevant Test Cases:	

Phasellus malesuada velit sed tortor ultrices sit amet convallis enim commodo. Curabitur velit urna, auctor vitae bibendum a, adipiscing sit amet dui. Duis sit

Figure 7. Requirement document with requirement name and serial number

Named and Numbered Test Case

Test Open App	
Test Case Name:	Test Open App
Test Case Serial Number:	TC_001

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc et dapibus velit. Fusce eu quam non est convallis faucibus eget eu eros. Pellentesque nec sem ut risus cursus lobortis. Nulla facilisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aliquam pellentesque turpis fermentum mi pulvinar auctor dapibus eget lectus. Fusce molestie sagittis nunc, sit amet condimentum nulla vestibulum a. Maecenas felis urna, laoreet eu feugiat at, pharetra nec velit. Phasellus eu risus magna. Sed vel risus magna. Etiam erat augue, posuere ac egestas vel, bibendum sit amet lectus.

Figure 8. Test case document with test case name and serial number

Step 3 - Bookmarks

Using the bookmark feature in word you will create a bookmark for each requirement and test case where you have a table and a serial number. The serial number becomes the bookmark name.

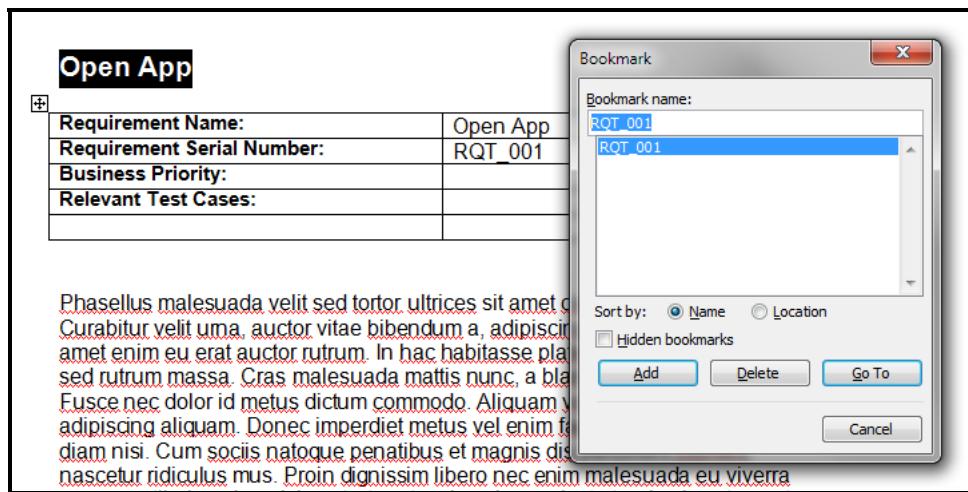


Figure 9. Bookmarking a requirement

Do the same for test cases.

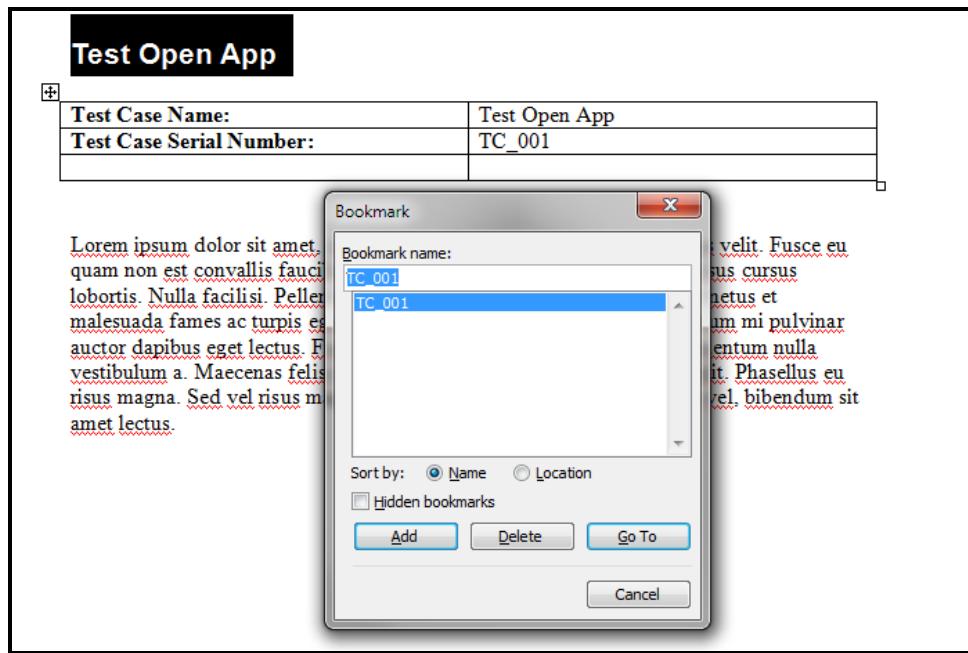


Figure 10. Bookmark a test case

Step 4 - Link Test Cases to Requirements Using Hyperlinks

VERY IMPORTANT DETAIL: Place both sets of documents in the same directory before you start hyperlinking. This way the documents are linked with a relative path and can later be moved anywhere as long as both the requirements and test cases documents stay in the same directory.

There are many benefits to hyperlinking between the documents. For example, if you are reading the requirement document and want to review the tests for a specific requirement simply click on the hyperlinked test case serial number and the test case document will be launched then positioned at the test case for easy review.

To link a test case to a requirement enter the requirement serial number in a blank cell then create a hyper link from that serial number to the bookmark in the requirement document.

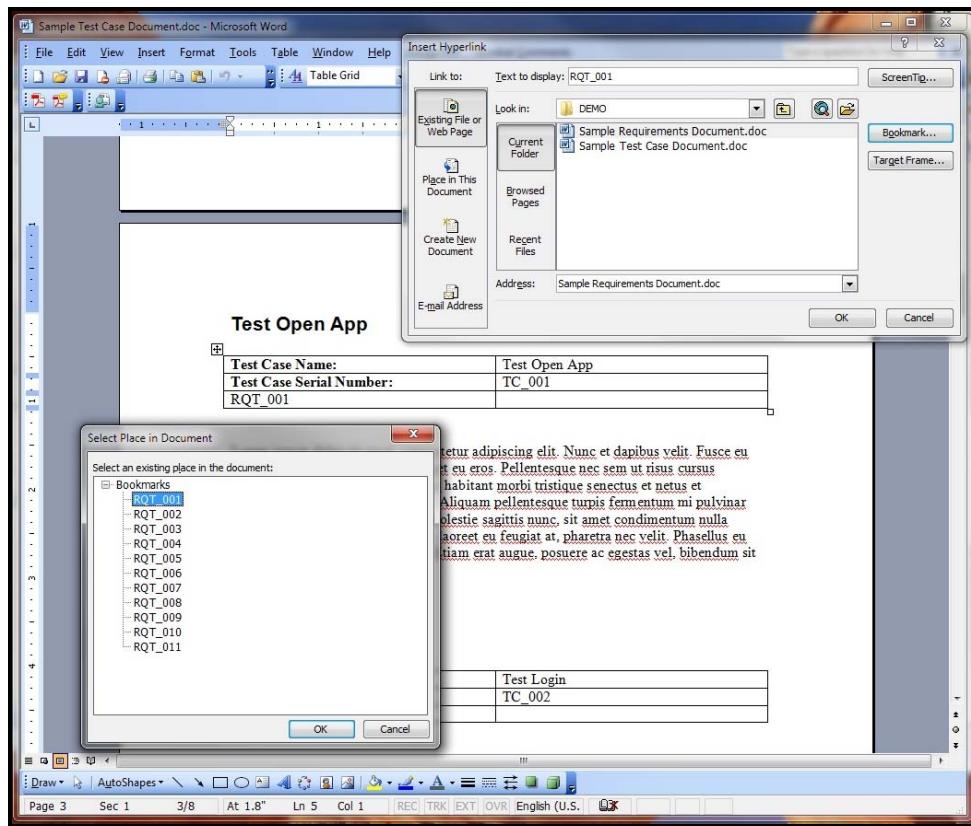


Figure 11. Linking a test case to a requirement

Reverse the process for requirements.

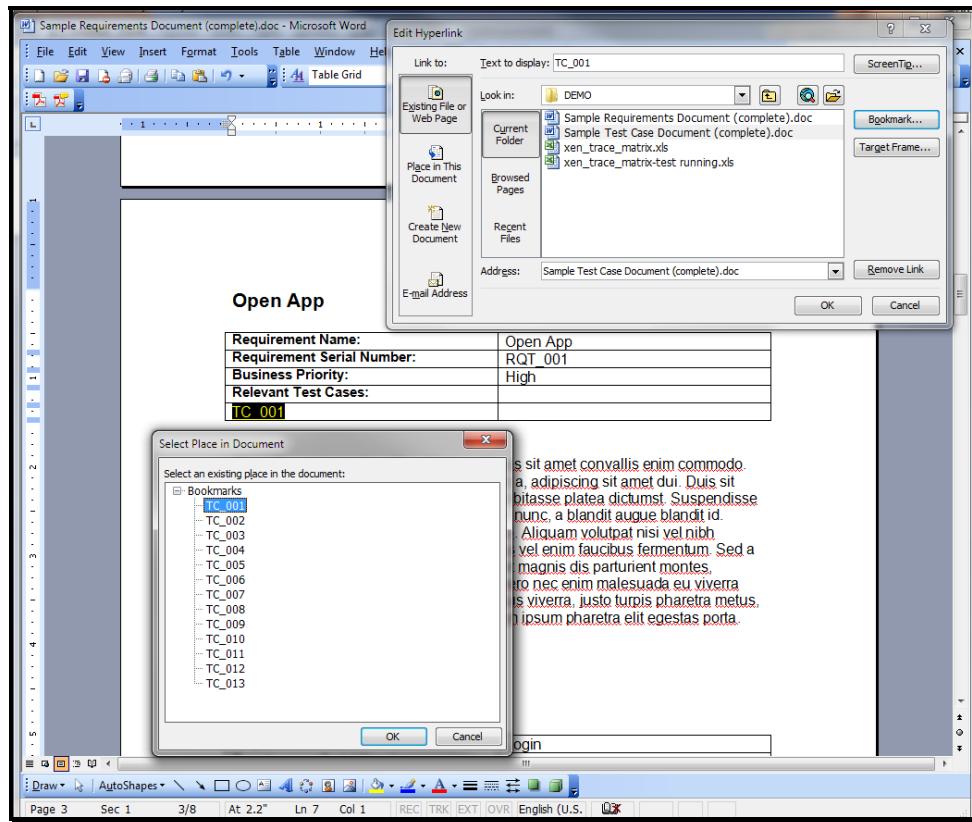


Figure 12. Linking a requirement to a test case

Step 5 - Assign Business Priority

Next you will need to assign a business priority to each requirement. It's best to work with the business end user for this task so that their priorities are represented in each requirement. If the business end user cannot participate then take your best estimation of the priority of each requirement.

Your requirements tables should look as follows:

Open App	
Requirement Name:	Open App
Requirement Serial Number:	RQT_001
Business Priority:	High
Relevant Test Cases:	
	TC 001

Phasellus malesuada velit sed tortor ultrices sit amet convallis enim commodo.
Curabitur velit urna, auctor vitae bibendum a, adipiscing sit amet dui. Duis sit

Figure 13. Assigning a business priority

A compelling byproduct of this business priority assignment effort is that the exercise gives the business user a chance to easily review all of the test cases associated with requirements. You and the business end user may find that some of the test cases were not testing the correct things and modifications are in order. Nobody had taken the time to review the correctness of the test cases because that review was too difficult without the RTM. It's real easy when you can simply hyperlink from a requirement to the associated test cases.

Step 6 - Review coverage

Review all of your requirements documents and all of your test case documents looking for tables without a hyperlink to the opposite document.

Requirements without links back to a test case indicates a lack of test coverage and the need to create more tests. Test cases without links back to a requirement potentially indicates wasted testing effort.

Some requirements may have several test cases to validate coverage. This condition makes sense - multiple tests to validate one requirement. But if you have a too many test cases per requirement, you will have wasted effort and, perhaps, a significant problem.

This example shows the one-to-many example of several test cases validating one requirement.

Enter Customers	
Requirement Name:	Enter Customers
Requirement Serial Number:	RQT_005
Business Priority:	Medium
Relevant Test Cases:	
TC_005	TC_006
TC_007	

Figure 14. Many test cases validate a single requirement

Step 7 - Run the Excel macro

Once you have your requirements documents linked to your test case documents then you can run the macro code included in Appendix A to pull the information out of the requirements documents into your multi-purpose RTM spreadsheet.

Figure 15. RTM after running macro

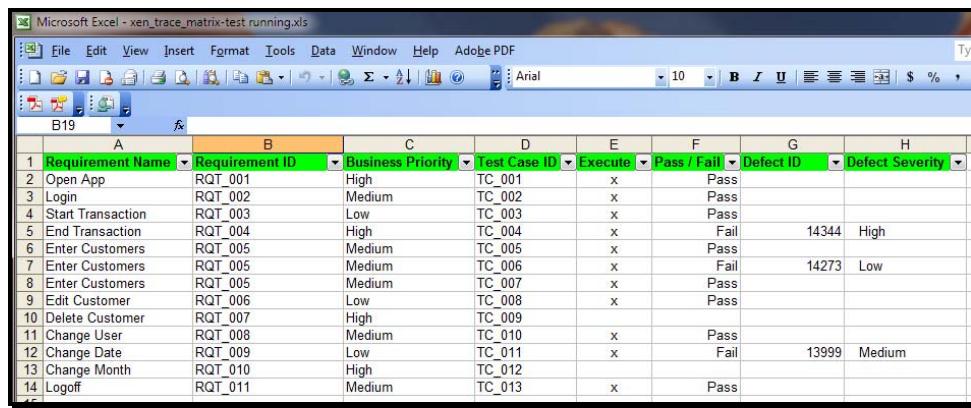
RTM Multi-purpose Spreadsheet

The spreadsheet, once populated and used for test management, has nearly magical qualities for something so simple while having only eight columns. When things are simple, they are useful. The AutoFilter feature makes simple and useful possible.

The magic is instant information and information is power. You now have the power to gain complete control over the project.

The first and foremost magical feature is that you can report test progress and test results in terms of requirements passed, failed, and not tested. All of your test reporting and test planning is now requirements-centric.

The following screen capture shows a simplified test in progress.



Requirement Name	Requirement ID	Business Priority	Test Case ID	Execute	Pass / Fail	Defect ID	Defect Severity
Open App	RQT_001	High	TC_001	x	Pass		
Login	RQT_002	Medium	TC_002	x	Pass		
Start Transaction	RQT_003	Low	TC_003	x	Pass		
End Transaction	RQT_004	High	TC_004	x	Fail	14344	High
Enter Customers	RQT_005	Medium	TC_005	x	Pass		
Enter Customers	RQT_005	Medium	TC_006	x	Fail	14273	Low
Enter Customers	RQT_005	Medium	TC_007	x	Pass		
Edit Customer	RQT_006	Low	TC_008	x	Pass		
Delete Customer	RQT_007	High	TC_009				
Change User	RQT_008	Medium	TC_010	x	Pass		
Change Date	RQT_009	Low	TC_011	x	Fail	13999	Medium
Change Month	RQT_010	High	TC_012				
Logoff	RQT_011	Medium	TC_013	x	Pass		

Figure 16. Test in progress using RTM to manage and report

The AutoFilter feature used on one or more columns allows you to create actionable and powerful reports on test progress. Here are a few questions that can be answered using the AutoFilter feature:

- List all high priority business requirements that have been tested and passed
- List all high priority business requirements that have been tested and failed
- List all high priority business requirements that have not been tested
- List the defect ID's created for all failed requirements sorted by business priority
- List all of the test cases that need to be executed to validate all high priority business requirements
- List all of the test cases that are planned to be executed this cycle
- List all of the high priority business requirements with an associated critical severity defect outstanding

The list of actionable information that can be generated from this spreadsheet is quite vast. By adding columns you can create a test management and reporting solution that meets your specific customized needs.

The RTM/test management spreadsheet has now given you great power to manage testing. Take the previously mentioned functional risk-based prioritization. With just a simple filter on the 'Business Priority' column in the spreadsheet you can find all test

cases that should be executed for all high priority business requirements. In a time crunch you know exactly which test cases to run.

Provisioning Checklist

Use the following list to plan your project before getting started.

- Plan the columns in the RTM/test management spreadsheet
- Plan the table in the requirements document to collect the information for the columns in your RTM
- Plan the naming convention for your requirement names
- Plan the numbering convention for your requirement serial numbers
- Plan the table in the test case documents
- Plan the naming convention for your test case names
- Plan the numbering convention for your test case numbers
- Plan the directory structure to contain both your requirement documents and your test case documents

Order of Events

Here is the order of events you should follow to implement this traceability strategy.

- Plan first using the Provisioning Checklist
- Collect all requirements documents and test case documents the place them in the same directory folder
- Add tables in the necessary places in the requirements documents and in the test case documents
- Go through each document filling in the name and serial for the requirement and test case in the respective tables
- Go through each document and create bookmarks for each requirement and test case serial number
- Hyperlink the test case document to the requirement document
- Hyperlink the requirement document to the test case document
- Review the requirement document for adequate test coverage. Add test cases where necessary
- Review the test case document to ensure all test cases are linked to at least one requirement
- Work with the business end user to assign business priority to each requirement
- Run the Excel macro to populate the multi-purpose RTM and test planning spreadsheet
- Perform test planning
- Perform test execution
- Perform test results reporting

Conclusion

We've created what we hope is an entertaining scenario in order to get the concept and code for this very powerful strategy of creating requirements traceability to test cases into your hands and hope many problems in testing can be solved with such a simple yet powerful solution.

The technique shown here for creating traceability works equally well for a mixture of Microsoft document types such as when requirements are stored in Word and test cases are stored in Excel. You will have to write your own macro code for this situation and maybe someone can post an article with their solution and code in the future.

Probably the biggest value of this exercise is that all of your test reporting is now in terms of requirements validated. Saying you have run 1000 test cases doesn't say much. Saying we ran 1000 test cases and validated 250 out of 300 high priority requirements provides information about how much of the application has been tested.

Appendix A

Excel Macro Code

This macro code is known to work with the sample documents on the xensight.com web site. You are free to use as needed but maintenance is all your responsibility and we provide no technical support.

```
Sub Get_Word_Tables()

' Variables used for the Excel worksheets

Dim strPath As String
Dim strDoc As String
Dim strPathFile As String

Dim strRqtID As String
Dim strRqtName As String
Dim strBusPri As String
Dim strTCID As String

Dim intRow As Integer ' Row of the spreadsheet for each record to be written
Dim k As Integer      ' Counter to increment through list of documents on "List of UC Docs" worksheet

' Word Objects
Dim wdApp As Word.Application
Dim wdDoc As Word.Document
Dim wdRange As Word.Range
Set wdApp = New Word.Application

' Variables used in accessing the Word document

Dim docActive As Document
Dim tblTrace As Table

Dim i As Integer
Dim j As Integer
Dim intTableCount As Integer
Dim intLength As Integer
Dim IngRow As Long
Dim IngCol As Long

Dim intRowsToProcess As Integer

' Start writing records in the "Master Summary" worksheet in row 2
intRow = 2

' Start in row 2 then go to each row with a document name. If you have 10 rqt documents then
' you start in row 2 and go through row 11 :::: For k = 2 to 11

For k = 2 To 2    ' The list of documents is hard coded since there are a finite know number of documents

    ' Show worksheet with list of documents
    Worksheets("List of RQT Docs").Activate
```

```
' Create string that is the path and file name of the document
' we need to open
strPath = Worksheets("List of RQT Docs").Range("A" & k).Value
strDoc = Worksheets("List of RQT Docs").Range("B" & k).Value
strPathFile = strPath & "\" & strDoc

' Open Word (the application)
wdApp.Visible = True

' Open a document in Word
wdApp.Documents.Open (strPathFile)

Set docActive = wdApp.ActiveDocument

intTableCount = wdApp.ActiveDocument.Tables.Count

Worksheets("Master Summary").Activate

For i = 1 To intTableCount
    If Left(wdApp.ActiveDocument.Tables(i).Cell(1, 1).Range.Text, 17) = "Requirement Name:" Then
        ' Trim off the cell terminator characters and write the Requirement Name
        strRqtID = Get_Cell_Contents(wdApp.ActiveDocument.Tables(i).Cell(1, 2).Range.Text)
        Worksheets("Master Summary").Range("A" & intRow).Value = strRqtID

        'Trim off the cell terminator characters and write the Requirement Serial Number
        strRqtName = Get_Cell_Contents(wdApp.ActiveDocument.Tables(i).Cell(2, 2).Range.Text)
        Worksheets("Master Summary").Range("B" & intRow).Value = strRqtName

        'Trim off the cell terminator characters and write the Business Priority
        strBusPri = Get_Cell_Contents(wdApp.ActiveDocument.Tables(i).Cell(3, 2).Range.Text)
        Worksheets("Master Summary").Range("C" & intRow).Value = strBusPri

        ' Write the Test Case ID's in the table by determining how many rows are in the table
        ' and getting the contents of row 5 through the end of the table

        intRowsToProcess = (wdApp.ActiveDocument.Tables(i).Rows.Count - 4)

        For j = 5 To (intRowsToProcess + 4)
            'Trim off the cell terminator characters and write the Test Case ID's or other text
            '*** There should always be an entry in row 4 column 1 so no inspection for empty is required
            strTCID = Get_Cell_Contents(wdApp.ActiveDocument.Tables(i).Cell(j, 1).Range.Text)
            Worksheets("Master Summary").Range("D" & intRow).Value = strTCID

            ' Need to write the other cells also.....
            Worksheets("Master Summary").Range("A" & intRow).Value = strRqtID
            Worksheets("Master Summary").Range("B" & intRow).Value = strRqtName
            Worksheets("Master Summary").Range("C" & intRow).Value = strBusPri

            ' Check entry in column 2
            strTCID = Get_Cell_Contents(wdApp.ActiveDocument.Tables(i).Cell(j, 2).Range.Text)

            If Len(strTCID) > 1 Then
                ' Next row in the spreadsheet if an entry exists in row 7 column 2
                intRow = intRow + 1

                Worksheets("Master Summary").Range("D" & intRow).Value = strTCID
```

```
' Need to write the other cells also....  
Worksheets("Master Summary").Range("A" & intRow).Value = strRqtID  
Worksheets("Master Summary").Range("B" & intRow).Value = strRqtName  
Worksheets("Master Summary").Range("C" & intRow).Value = strBusPri  
  
Else  
    ' Do Nothing, the cell is empty  
End If  
  
' Next row in the spreadsheet  
intRow = intRow + 1  
Next j  
End If  
Next i  
  
' Close the document since the values from the tables have been captured  
wdApp.Documents.Close  
  
Next k  
  
' Close Word  
wdApp.Visible = False  
  
Set wdApp = Nothing  
  
End Sub  
  
Function Get_Cell_Contents(strCell As String)  
  
'This function trims off the cell terminator characters and return the cell's content  
' strCell = workbook cell  
  
Dim intLength As Integer  
  
intLength = Len(strCell) - 2  
Get_Cell_Contents = Trim(Left(strCell, intLength))  
  
End Function
```

Appendix B

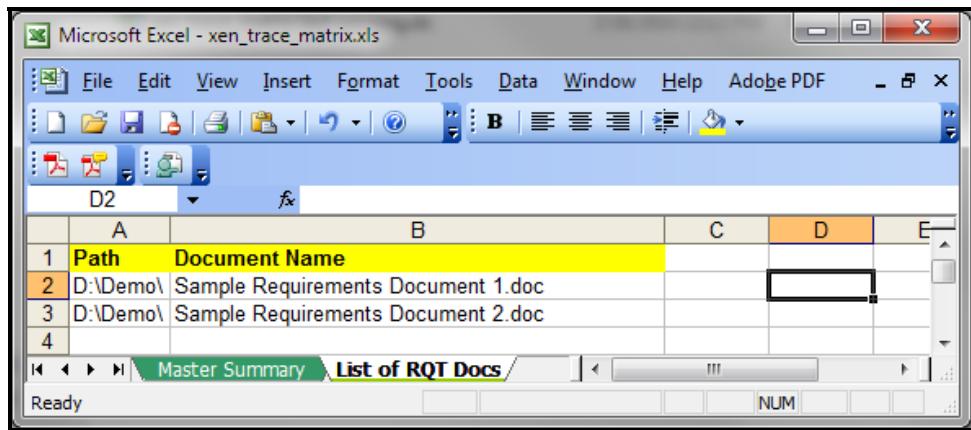
How the code works

The layouts of the spreadsheet and tables in the sample document screen captures is important and are tightly coupled to the code.

Spreadsheet

The spreadsheet should have two tabs, "Master Summary" and "List of RQT Docs." The Excel macro reads from or writes to the tabs according to the name on the tab. If you change the name on the tab then you will need to modify the code.

Here is a screen capture of the spreadsheet:



Here is code that refers to the content on the specific tab:

```
' Show worksheet with list of documents
Worksheets("List of RQT Docs").Activate

' Create string that is the path and file name of the document
' we need to open
strPath = Worksheets("List of RQT Docs").Range("A" & k).Value
strDoc = Worksheets("List of RQT Docs").Range("B" & k).Value
strPathFile = strPath & "\" & strDoc
```

The "List of RQT Docs" tab should contain the path and file name of all of your requirements documents that you want to include in the traceability matrix.

The macro has the number of documents hard coded. If you add documents then you will need to change the code to accommodate the number of documents in your list.

Here is the code that needs to be altered:

```
' Start in row 2 then go to each row with a document name. If you have 10 rqt documents then
```

' you start in row 2 and go through row 11 :::: For k = 2 to 11

For k = 2 To 3 ' The list of documents is hard coded since there are a finite know number of documents

Table Format in Requirement Document

The table format in the requirement document is important. If you change the format or the text in the left cells then you will need to modify the code.

Requirement Name:	Open App
Requirement Serial Number:	RQT_001
Business Priority:	High
Relevant Test Cases:	
TC_001	

The macro code recognizes that your existing requirement document most likely already contains tables. The macro loops through the tables collection in the document object and inspects each table for the text string "Requirement Name:" in row 1 column 1 of the table. If the text string is found then the table is processed into the trace matrix.

Here is the specific line of code that inspects row 1 column 1:

```
If Left(wdApp.ActiveDocument.Tables(i).Cell(1, 1).Range.Text, 17) = "Requirement Name:" Then
```

If you change the format of the table then you will need to change the macro code accordingly.

Multiple Test Case Links

The macro automatically accommodates multiple test case links in a table and will find all links as long as you place one link per cell as shown here:

Requirement Name:	Open App
Requirement Serial Number:	RQT_001
Business Priority:	High
Relevant Test Cases:	
TC_001	TC_042
TC_044	TC_051
TC_052	

Add as many rows of test case links as you need.