

Establishing Traceability of Software Test Coverage Using MS-Excel – A Simple Approach

- Venkat Moncompu[±]

Abstract: Confidence in test coverage comes with traceability of test artifacts to requirements. Traceability also provides the capability of regression test selection for interim (and final) software releases. While there are sophisticated tools that provide these capabilities, this article outlines a simple creation, analysis and reporting capability for traceability using MS-Excel and MS-Query. MS-Query comes standard with the Microsoft Office suite of applications.

Traceability: The ability to associate/look-up related artifacts upstream and downstream of the testing phase in the software development process. References to upstream process may be references to artifacts from the tasks/activities that precede the testing phase in timeline such as mapping test cases to requirements. References to downstream process may be references to defects created by test cases and identification of surround test cases for regression around the specific defect based on related requirements and/or features. The combination of excel spreadsheets, MS query and pivot tables provide a powerful means to capture these traceability creation, analysis & reporting needs.

Assumptions:

In the proposed example template, the following assumptions are made:

1. There cannot be a requirement without a feature. Feature is something the business (user) wants while requirement is a verifiable contract to meet that stated feature (or user need) or part of that feature.
2. Ad-hoc tests (be it UAT or System tests) need not correspond to any specific requirement but will be related to one of the features.
3. A scenario is a unique path of execution through a specific use-case. A use-case is a generalization of all the possible execution flows toward a specific goal.
4. A single requirement has to map to at least one feature/user need. It can also map to partial features.
5. A single test scenario can map to one or more requirements.
6. For every feature, there has to be at least one requirement.

Using MS-Query is a very powerful way to create complex traceability reports using SQL queries. Setting up the data source requires specific tab and cell ranges naming conventions to be followed for the queries developed in MS-Query to work. The following describes the setup procedure for the excel spreadsheet database as an external data source using MS-Query.

1. Business analysts provide the feature to requirement traceability through the generation of requirement documents. Testers either verify that this is accurate and/or identify the primary and secondary requirements to the features from these documents during the review process. This is captured in the “FeatureReqMap” tab and associated table name given to the range of cells in the tab.

	A	B	C	D
1	FeatureID	PrimaryReq	SecondaryReq	TertiaryReq
2	Feat ID 1	Req-2	Req-10	
3	Feat ID 2	Req-1	Req-3	Req-4
4	Feat ID 3	Req-7		
5	Feat ID 4	Req-8		
6	Feat ID 5	Req-5	Req-6	Req-11
7	Feat ID 6	Req-6	Req-10	Req-9
8	Feat ID 7	Req-6		
9	Feat ID 8	Req-8	Req-3	
10	Feat ID 9	Req-9		
11	Feat ID 10	Req-10	Req-11	Req-4
12	Feat ID 11	Req-1	Req-15	
13	Feat ID 12	Req-11		
14	Feat ID 13	Req-13		
15	Feat ID 14	Req-14	Req-15	
16	Feat ID 15	Req-12	Req-9	
17	Feat ID 16	Req-16		

Figure 1 - Feature to Requirement mapping

2. "MasterDefectTable" is the name of the tab and associated named cells range for the relation established between reported defects – QCIDs, the corresponding requirement by their ID and the feature it is required to meet. This is the master table of defects mapped to test cases.

	A	B	C	D	E	F	G	H	I	J
1	DefectID	Summary	QAScenario	UATID	UATScenario	UseCaseID	ReqID	Comment	Features	DefectSeverity
2	QCID-1	Summary for QCID 1	QA Scenario 1124				Req-10		Feat ID 10	High
3	QCID-2	Summary for QCID 2	QA Scenario 1886	UAT ID 1	UAT Scenario 1	Use Case ID 11	Req-10		Feat ID 10	Medium
4			QA Scenario 1002					Ad-hoc QA Test		
5	QCID-4	Summary for QCID 4	QA Scenario 1002	UAT ID 3	UAT Scenario 3	Use Case ID 2	Req-9		Feat ID 9	High
6			QA Scenario 1002	UAT ID 3	UAT Scenario 3	Use Case ID 1	Req-8		Feat ID 8	
7	QCID-6	Summary for QCID 6	QA Scenario 1445					Ad-hoc QA Test	Feat ID 4	low
8	QCID-7	Summary for QCID 7	QA Scenario 1345			Use Case ID 10	Req-4		Feat ID 5	Medium
9	QCID-8	Summary for QCID 8		UAT ID 3	UAT Scenario 3		Req-1		Feat ID 1	High
10	QCID-9	Summary for QCID 9		UAT ID 2	UAT Scenario 2		Req-2		Feat ID 2	High
11	QCID-10	Summary for QCID 10	QA Scenario 1243			Use Case ID 6	Req-1		Feat ID 2	Medium
12	QCID-3			UAT ID 5	UAT Scenario 5			Ad-hoc UAT test	Feat ID 7	low
13	QCID-14			UAT ID 6	UAT Scenario 6			Ad-hoc UAT test	Feat ID 11	Medium
14			QA Scenario 1154			Use Case ID 3	Req-3		Feat ID 8	
15			QA Scenario 1005	UAT ID 3	UAT Scenario 3	Use Case ID 2	Req-13		Feat ID 13	
16			QA Scenario 1203			Use Case ID 10	Req-5		Feat ID 5	
17			QA Scenario 1105				Req-6		Feat ID 6	
18	QCID-5	Summary for QCID 5	QA Scenario 1100	UAT ID 4	UAT scenario 4		Req-7		Feat ID 3	High

Figure 2 - Master Defect Table

3. The “ReqTCMap” tab and table cell range name given would be the test cases generated to verify each of the requirements. Maintained by the QC leads or test case reviewers, it establishes the test cases created for each requirement.

	A	B
1	ReqID	TestCases
2	Req-1	Test case 1, Test case 2, Test case 3, Test case 4, Test case 5, Test case 6
3	Req-2	test case 10, Test case 11, Test case 7, Test case 8, Test case 9
4	Req-3	Test case 12, Test case 13, Test case 14
5	Req-4	Test case 20, Test case 21
6	Req-5	Test case 15, Test case 16, Test case 17, Test case 18, Test case 19
7	Req-6	Test case 1, Test case 2, Test case 6
8	Req-7	Test case 3, Test case 11
9	Req-8	Test case 4, Test case 17, Test case 18
10	Req-9	Test case 2, Test case 5
11	Req-10	Test case 10
12	Req-11	Test case 11, Test case 12
13	Req-12	Test case 12, Test case 13
14	Req-13	Test case 12, Test case 13, Test case 16
15	Req-14	Test case 13, Test case 14, Test case 15
16	Req-15	Test case 15, Test case 16, Test case 17, Test case 18
17	Req-16	Test case 8

MasterDefectTable FeatureReqMap ReqTCMap Comments Reports

Ready Scroll Lock

Figure 3 – Test Cases Created for the Defined Requirements

4. “Reports” tab is where the generated traceability reports are finally pasted. Here is the result set of query 1 as described in the reporting section below.

Given a defect id, to find:			
1. the corresponding requirement it was reportedly mapped to.			
2. the other test cases associated with the specific requirement.			
3. the features upstream of the development lifecycle.			
DefectID	ReqID	TestCases	Features
QCID-1	Req-10	Test case 10	Feat ID 10
QCID-2	Req-10	Test case 10	Feat ID 10
QCID-4	Req-9	Test case 2, Test case 5	Feat ID 9
QCID-7	Req-4	Test case 20, Test case 21	Feat ID 5
QCID-8	Req-1	Test case 1, Test case 2, Test case 3, Test case 4, Test case 5, Test case 6	Feat ID 1
QCID-9	Req-2	test case 10, Test case 11, Test case 7, Test case 8, Test case 9	Feat ID 2
QCID-10	Req-1	Test case 1, Test case 2, Test case 3, Test case 4, Test case 5, Test case 6	Feat ID 2
QCID-5	Req-7	Test case 3, Test case 11	Feat ID 3

Figure 4 - Regression Test Cases from Traceability

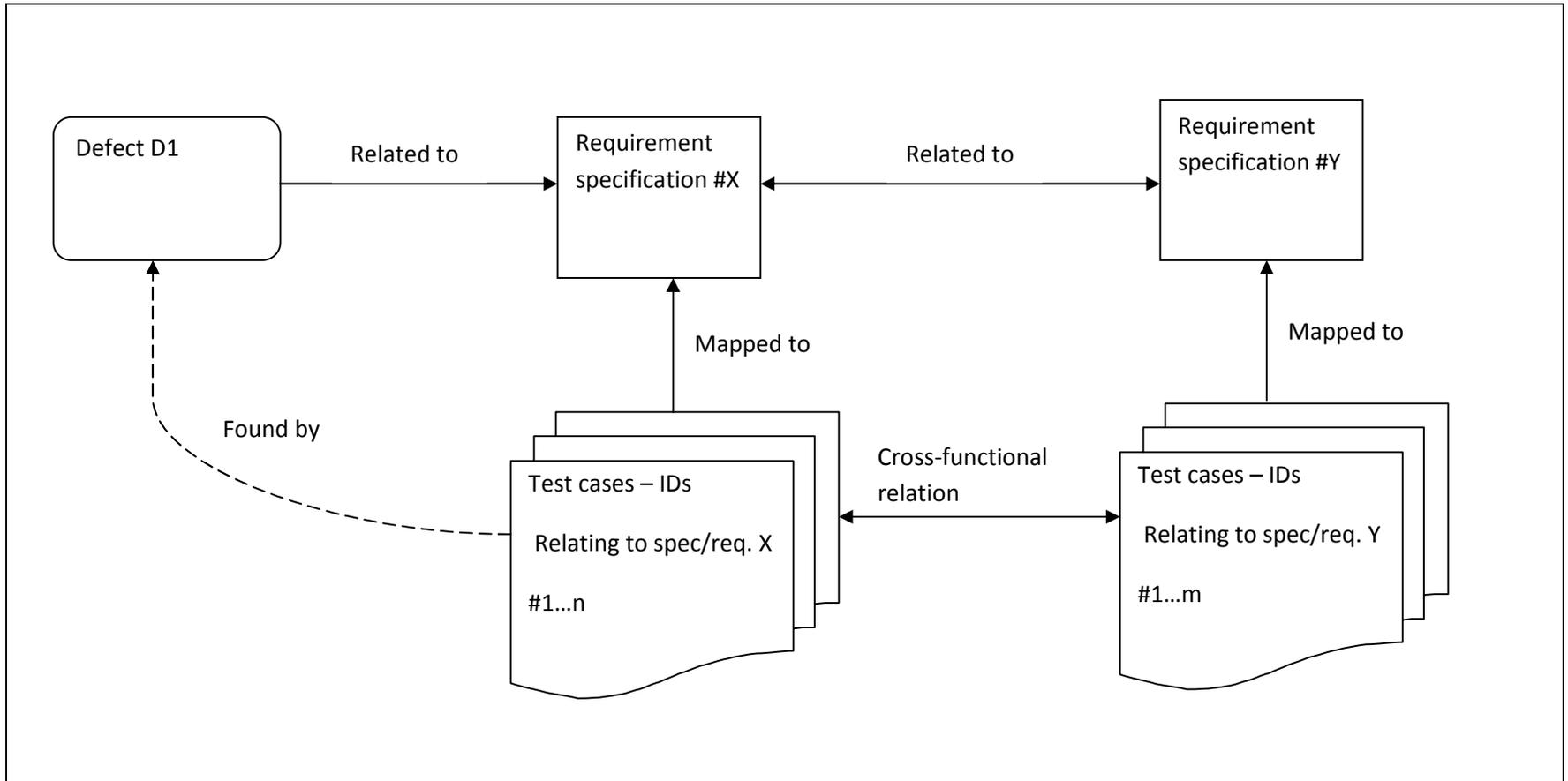


Fig 1: Traceability Map of Defects to Test cases via Requirements.

Given the situation in fig. 1, the goal is for the user to be able to find:

1. All test cases (and scenarios) 1...n relating to the requirement X that was mapped to the specific defect D1?
2. The defect D1 may or may not have been found by one of the test cases 1...n; if so, where is the gap – is it in the requirement or is it in the mapping of the test cases (or scenarios) to the requirement X? Otherwise, find all test cases relating to requirement #X.
3. If the defect is a side-effect or related to a cross-functional relationship of requirement X with another requirement Y, then the tool should identify the related scenarios from Test set 1...m via the requirement relationship to identify the “surround” test scenarios that have to be run as part of regression test for defect fix D1.

Reporting:

Having set-up the excel spreadsheet data-source in this manner. There are many ways to use the traceability spreadsheet with MS-Query. The only limitation is that MS-Query requires the excel data-source to be closed while working with queries and a temporary excel work file to return the results of the query to. We've used report1.xlsx as this temporary file until after all the queries are executed and finally, copied (& pasted) into the final "Reports" tab of the traceability matrix spreadsheet.

1. Find all the test cases that relate to a feature associated with a defect reported:

To identify all surround test cases related to a requirement that is mapped to a defect reported, the following query will generate all related test cases. The resulting test cases will give a regression suite of tests to run for the specific defect in question. {Query is saved in a file named "SurroundTestQuery.dqy"}

```
SELECT MasterDefectTable.DefectID, ReqTCMap.TestCases
FROM MasterDefectTable MasterDefectTable, ReqTCMap ReqTCMap
WHERE MasterDefectTable.ReqID = ReqTCMap.ReqID AND ((ReqTCMap.TestCases Is Not Null) AND
(MasterDefectTable.DefectID Is Not Null))
```

2. Find all the test cases based on the features that ad-hoc tests relate to since these ad-hoc tests may not necessarily map to explicit requirements:

For ad-hoc tests that are run by QA or by UAT, there usually will not be an associated requirement. In that case, the feature id is used to identify all of the related requirements for which a larger sub-set of test cases may be run for "surround" testing coverage. The following query helps list out these tests. {Query is saved in a file named "AllTestCasesFromAdHocTests.dqy"}

```
SELECT MasterDefectTable.DefectID, ReqTCMap.ReqID, ReqTCMap.TestCases
FROM FeatureReqMap FeatureReqMap, MasterDefectTable MasterDefectTable, ReqTCMap ReqTCMap
WHERE FeatureReqMap.PrimaryReq = ReqTCMap.ReqID AND MasterDefectTable.Features =
FeatureReqMap.FeatureID AND ((MasterDefectTable.DefectID Is Not Null) AND
(MasterDefectTable.ReqID Is Null))
```

3. Find all test cases based on a secondary requirement for a specific feature identified for a primary requirement for which a specific report is identified. This is the situation as described in fig. 1 above where the user is expected to find all test cases corresponding to the requirement Y which is related to requirement X through cross-functional relationship. {Query is saved in a file named "SecondaryTCsfromDefectID&Surround.dqy"}

```
SELECT SurroundTestQuery.DefectID, ReqTCMap.ReqID, ReqTCMap.TestCases
FROM FeatureReqMap FeatureReqMap, MasterDefectTable MasterDefectTable, ReqTCMap ReqTCMap,
SurroundTestQuery SurroundTestQuery
WHERE SurroundTestQuery.DefectID = MasterDefectTable.DefectID AND MasterDefectTable.Features =
FeatureReqMap.FeatureID AND FeatureReqMap.SecondaryReq = ReqTCMap.ReqID
ORDER BY SurroundTestQuery.DefectID
```

The advantage of using MS-Excel is that the results from the first query can be used as a data source to match against the matching secondary requirements by features. And by extension, to the any level (tertiary or quaternary...) of interdependency of cross-functional influence, the same exercise may be carried out to tertiary and quaternary order of requirement mapping and reporting.

Conclusion:

There are many advantages of using MS-Query with MS-Excel. Perhaps, the biggest advantage is that it is dynamically updateable without manually editing the report tables. The original spreadsheet data-source can be independently maintained and updated as long as the minimal nomenclature restrictions are followed. Therefore the maintenance overhead is low. Further, the pivot table so generated from the set of queries run can be plugged right back into an Excel report where filters can be used to easily view the extent of test coverage required to verify the regression test of a specific defect fix.

The flexibility of querying based on multiple data fields opens up enormous possibilities for reporting by defect severity against specific feature, identifying gaps in requirements by feature etc. A more detailed benefits and usage of a traceability tool in the larger context of the software development process can be had from a previous article by the author published in Software Test & Performance magazine titled "Role of Artifacts in QA" dated Sep. 2008.

Appendix:

Files referenced are

1. Report1.xlsx – as a temporary working file for MS-Query results set pivot table.
2. Master database or data-source file – SampleTraceV1_0.xlsx as described above.
3. The four (4th being the tertiary order of requirement mapping as described in Fig.1 above) queries used for the different reports named (dqy extension for database query file):
AllTestCasesFromAdHocTests.dqy, SurroundTestQuery.dqy,
SecondaryTCsfromDefectID&Surround.dqy and TertiaryTCsfromDefectID&Surround.dqy.

About the author:

Venkat Moncompu[±] is a senior quality analyst with Deloitte Touche Tohmatsu, Chicago. He has 14 years of IT industry experience with an M.S. degree in Engineering from Arizona State University and a Masters Certificate in Project Management from Villanova University. He can be reached at venkatams {at} yahoo {dot} com