Quality Excellence for Suppliers of Telecommunications Forum (QuEST Forum)

TL 9000 Quality Management System

Measurements Handbook

SFQ Examples

8.1 SFQ Examples

8.1.1 – SFQ Example

The following example illustrates calculation of the Software Fix Quality measurement. Software fixes are counted regardless of the method used to package/deliver the fix. For this example, release numbers are used to identify the packages of released fixes.

An organization has three active releases, 1.0, 1.1, and 1.2. Software fixes have been made available as shown in the left side of Table 8.1.1-1. The right side of the table shows the number of software fixes reported for the Software Fix Quality measurement in the reporting months shown.

Software Fixes Released				Reporting Month					
Release		Release Date	Fix Cnt	Jan '17	Feb '17	Mar '17	Apr '17	May '17	Jun '17
1.0	1.0	Dec-15	n/a						
	1.0.1	Jan-16	6						
	1.0.2	Apr-16	1	1	1	1			
	1.0.3	Jun-16	5	5	5	5	5	5	
1.1	1.1	Jun-16	15	15	15	15	15	15	
	1.1.1	Dec-16	4	4	4	4	4	4	4
	1.1.2	Jun-17	2						2
1.2	1.2	Jan-17	10	10	10	10	10	10	10
	1.2.1	Mar-17	5			5	5	5	5
	1.2.2	Jun-17	1						1
Software Fixes Counted in the Reporting Month				**35**	**35**	**40**	**39**	**39**	**22**

Table 8.1.1-1 Example SFQ Release/Fix Data

In the January through June timeframe, the following problems with the delivered fixes are identified:

1. In January 2017, during the installation of Release 1.1.1 it becomes evident that the release cannot be installed.
   - All 4 fixes delivered in the release are counted as defective fixes for January 2017.
2. In February 2017, a customer reports a fix in Release 1.1.1 did not fix the intended problem.
   - The fix is not counted as a defective fix since it was already reported as a defective fix in January 2017.

3. Also in February 2017 a Minor problem is reported because one of the fixes in Release 1.1 did not fix the intended problem.
   - The fix is counted as a defective fix in February since the fix did not correct the intended problem.
4. Also in February, a Major problem determined to be a side effect of another (different) fix provided in Release 1.1 is reported.
   - The fix is not counted as a defective fix in February because the date the fix is found defective is outside the 6-month window of the fix release date for Release 1.1.
5. In March 2017 a Major problem is reported because one of the fixes in Release 1.2 did not fix the intended problem.
   - The fix is counted as a defective fix in March.
6. In April 2017, additional internal testing by the organization for the fixes in Release 1.2.1 determines that one of the fixes doesn't completely correct the intended problem.
   - The fix is counted as a defective fix in April.
7. In May 2017, a Critical problem determined to be caused by a fix in Release 1.1.2 is reported.
   - The fix is counted as a defective fix in May.
8. Also in May 2017, a Minor problem determined to be a side effect of the second fix in Release 1.1.2 is reported.
   - The fix is not counted as a defective fix in May since the problem for the side effect is opened as a Minor problem.
9. In June 2017, a new customer coming up on Release 1.0.3 reports a Critical problem and cannot install the release.
   - Since the installation date is not reported within 12 months from the release date of the fixes, the 3 fixes provided in the release are not included in the June 2017 defective fix count.

The number of defective software fixes reported each month for the Software Fix Quality measurement would be determined as shown in Table 8.1.1-2.

Table 8.1.1-2 Example SFQ Monthly Defective Fixes

	Jan 2017	Feb 2017	Mar 2017	Apr 2017	May 2017	Jun 2017
Number of Defective Software Fixes						
Release 1.0	0	0	0	0	0	0
Release 1.1	4	1	0	0	1	0
Release 1.2	0	0	1	1	0	0
Total	4	1	0	1	1	0

The resulting monthly source data and measurement calculations (rounded to the nearest integer) are shown in Table 8.1.1-3.

Table 8.1.1-3 Example SFQ Monthly Source Data and Measurement Calculation

	Jan 2017	Feb 2017	Mar 2017	Apr 2017	May 2017	Jun 2017
Afactor (Fa)	12	12	12	12	12	12
Software Fixes (Fc)	35	35	40	39	39	22
Defective Fixes (DFc)	4	1	0	1	1	1
%Defective Per Yr	137%	34%	0%	31%	31%	55%

Note that, as shown in the January 2017 SFQ measurement calculation, the monthly calculation may exceed 100% due to the annualization factor being applied to the measurement.

For the month of June 2017, the TL 9000 SFQ data reported is shown in Table 8.1.1-4.

Table 8.1.1-4 SFQ Data Table Report

Identifier	Value
MeasurementID	SFQ
Fa	12
DFc	1
Fc	22

8.1.2 – Frequently Asked Questions

8.1.2.1 How do I count software fixes?

Organizations have one or more means by which software fixes are delivered or made available to the customer for implementation. These different types of delivery mechanisms include (but are not limited to) patches, files, maintenance releases, updates, dot releases, fix releases, etc. Although the actual implementation method differs, each of these means would include some type of notification of the availability of the software change and information on what fixes are included, such as a release letter or product bulletin.

Organizations can use the customer notification of the software change to obtain the number of fixes to be included in the SFQ measurement.

The information must be descriptive enough to ensure only fixes to problems requiring changes in the product software are counted. Fixes associated with paper documentation or enhancement requests would not be counted. The customer notification would need to include fixes to problems in the delivered software found by the organization as well as those found by customers; otherwise, these fixes would need to be identified and counted separately and the internal and customer fix counts added together to obtain the reported SFQ counts.

An alternate means of identifying fixes to be included in the SFQ measurement is by utilizing the organization's problem tracking tool to identify problems fixed. As with the previous method, the tool must be able to identify what release(s) the problem is being fixed in, distinguish between defects and enhancements as well as distinguishing between fixes requiring product software changes and those that do not (for example, paper documentation changes, third-party software changes). If separate tools are used to track customer problems and internally found problems, counts from the two systems would need to be added together to obtain the reported SFQ counts.

8.1.2.2 How can I use the SFQ Measurement?

The Software Fix Quality measurement is the percentage of software fixes determined to be defective. The higher the percentage, the greater is the risk that the installation of a software fix to correct a problem will either fail to correct it or else introduce additional problems into the network.

When using the SFQ measurement, especially to set goals and drive continuous improvement, it is important to consider the TL 9000 Performance Data Reports smoothing rules and use the smoothed averages. Monthly snapshots may demonstrate too much variability to provide an accurate representation of the Software Fix Quality trend due to the delay in the discovery of defective fixes. This variability will be more obvious in products that only release fixes a few times a year.

The SFQ measurement should trend downward, eventually reaching 0. Where this is not happening, organizations should consider performing defect analysis on the defective fixes to identify possible process improvements. The analysis should be focused on why the fix provided did not work, not on what caused the original problem. A detailed defect analysis should identify the root cause of the problem, and measures that could have detected the defective fix before it was made available and prevented it from being introduced into the software.

If there is a spike in the percentage of defective fixes, the organization could perform a high-level defect analysis to see if the problems are

related to a particular release, customer, product platform, etc. This could help identify possible areas for improvement at a lower cost than a more detailed analysis of individual defects.