

**Telecommunications Industry Association
Business Performance Community (TIA -BPC)**

**TL 9000
Quality Management System
Measurements Handbook**

SFQ Examples

**Telecommunications Industry Association (TIA)
Business Performance Community (BPC)**

**TL 9000
品質マネジメントシステム
測定法ハンドブック**

SFQ 計算例

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		FixCnt	Jan	Feb	Mar	Apr	May	Jun
	Date			'17	'17	'17	'17	'17	'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:

- 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.

8.1 SFQ 計算例

8.1.1 – SFQ 計算例

ソフトウェア問題処置品質測定法の計算例を以下に示す。ソフトウェアの問題処置は問題処置をパッケージして配布する方法の如何にかかわらず計数される。

例えば、リリース番号は、リリースされた問題処置のパッケージを識別するために用いられる。

ある組織には、現在運用中の 3 つのリリース、1.0、1.1、及び 1.2 がある。現在までに利用可能になっているソフトウェア問題処置リリースを表 8.1.1-1 の左側に示す。表の右側は、表示されている報告月のソフトウェア問題処置品質測定法で報告されたソフトウェア問題処置数を示す。

ソフトウェア問題処置リリース				報告月					
リリース	リリース日	問題処置計数		2017	2017	2017	2017	2017	2017
				年	年	年	年	年	年
				1月	2月	3月	4月	5月	6月
1.0	1.0	12月15日	適用外						
	1.0.1	1月16日	6						
	1.0.2	4月16日	1	1	1	1			
	1.0.3	6月16日	5	5	5	5	5	5	
1.1	1.1	6月16日	15	15	15	15	15	15	
	1.1.1	12月16日	4	4	4	4	4	4	4
	1.1.2	6月17日	2						2
1.2	1.2	1月17日	10	10	10	10	10	10	10
	1.2.1	3月17日	5			5	5	5	5
	1.2.2	6月17日	1						1
報告月におけるソフトウェア問題処置計数				35	35	40	39	39	22

表 8.1.1-1 例-SFQ のリリースと問題処置データ

1月から6月まで期間内で、以下の問題処置の実施が確認された。

- 2017年1月、リリース 1.1.1 のインストール中に、その問題処置はインストールできないことが明らかになった。

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.

- そのリリースで提供された4つ全ての問題処置は、2017年1月の欠陥のある問題処置として計数される。
2. 2017年2月、顧客から、リリース1.1.1での問題処置が意図した問題を処置しなかったと報告があった。
 - この問題処置は2017年1月に欠陥問題処置として既に報告されているので、欠陥問題処置とは計数されない。
 3. また、2017年2月に、リリース1.1の中の一つの問題処置が意図した問題を処置していなかったため、軽微な問題が報告された。
 - この問題処置は、意図した問題を修正していないので、欠陥問題処置として計数された。
 4. また、2月に、リリース1.1で提供された他の(異なる)問題処置の副作用となると判明した重大な問題が報告された。
 - この問題処置の日付がリリース1.1に対する問題処置のリリース日の6ヶ月の範囲外で発見されたため、この問題処置は2月の欠陥問題処置としては計数されない。
 5. 2017年3月に、リリース1.2の中の一つの問題処置が意図した問題を処置していないため、重大な問題が報告された。
 - この問題処置は3月の欠陥問題処置として計数される。
 6. 2017年4月に、組織によるリリース1.2.1の問題処置に対する追加内部試験によって、一つの問題処置が意図した問題を完全には処置していないことが判明した。
 - この問題処置は4月の欠陥問題処置として計数される。
 7. 2017年5月に、致命的な問題が、リリース1.1.2の問題処置に起因することが判明した。
 - この問題処置は5月の欠陥問題処置として計数される。
 8. また、2017年5月に、リリース1.1.2における二番目の問題処置の副作用であると判明した軽微な問題が報告された。
 - この副作用に関する問題は、軽微な問題として解決していないため、この問題処置は欠陥問題処置として計数されない。
 9. 2017年6月に、リリース1.0.3の新規顧客が致命的な問題を報告し、このリリースをインストールすることが出来なかった。
 - 問題処置のリリース日から12ヶ月以内に、インストール日付が報告されなかったため、このリリースへ提供された3つの問題処置は2017年6月の欠陥問題処置計数には含まれない。

ソフトウェア問題処置品質測定法のために、毎月報告される欠陥のあるソフトウェア問題数は、表8.1.1-2に示すようになる。

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	1	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	41	22
Defective Fixes (DFc)	4	1	1	1	1	1
%Defective Per Yr	137%	34%	30%	31%	29%	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.

表 8.1.1-2 例- SFQ の月毎の欠陥問題処置

	2017年 1月	2017年 2月	2017年 3月	2017年 4月	2017年 5月	2017年 6月
欠陥のあるソフトウェア問題処置数						
リリース 1.0	0	0	0	0	0	0
リリース 1.1	4	1	0	0	1	0
リリース 1.2	0	0	1	1	0	0
合計	4	1	1	1	1	0

月毎の元データと測定値計算結果 (整数に丸めてある)を表 8.1.1-3 に示す。

表 8.1.1-3 例- SFQ の月毎の元データと測定値計算結果

	2017年 1月	2017年 2月	2017年 3月	2017年 4月	2017年 5月	2017年 6月
年次換算係数 (Fa)	12	12	12	12	12	12
ソフトウェア問題処置数 (Fc)	35	35	40	39	41	22
欠陥問題処置数(DFc)	4	1	1	1	1	1
年間欠陥率 (%)	137%	34%	30%	31%	29%	55%

2017年1月のSFQ測定値計算結果に示す通り、月毎の測定値計算に年次換算係数を適用しているため、100%を越えていることに留意する。

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

2017年6月のTL 9000 SFQの報告データを表 8.1.1-4.に示す。

表 8.1.1-4 SFQ データ表報告

識別子	値
MeasurementID	SFQ
Fa	12
DFc	1
Fc	22

8.1.2 – よくある質問

8.1.2.1 ソフトウェア問題処置はどのように計算するか？

組織はソフトウェア問題処置を実施できるよう顧客に配布し、又は利用可能にする一つ以上の手段を持っている。これらの様々なタイプの配布の仕組みには、パッチ、ファイル、保守リリース、更新、ドットリリース、問題処置リリース等がある(これらに限定するものではない)。実際の実装方法は異なるが、これらの各手段には、ソフトウェア変更の可用性に関する何らかの種類の通知と、リリースレターや製品に関する公告などの問題処置が含まれている情報が含まれる。

組織は、SFQ 測定に含まれるべき問題処置数の情報を得るために顧客へのソフトウェアの変更に関する通知を利用することが出来る。情報は、その製品のソフトウェアの変更を必要とする問題への処置のみを計数することを確実にするのに十分な記述内容でなければならない。紙に書かれた文書に関連した問題処置や機能強化の要求は計数されることはない。顧客への通知には、組織だけでなく顧客によって発見された配布ソフトウェアの問題の処置を含んでいなければならない：そうでない場合は、それらの問題処置は、個別に識別され計数される必要があり、また報告する SFQ の計数値を得るためには、内部の問題処置と顧客の問題処置の計数値が合算される必要がある。

SFQ 測定に含まれる問題処置を識別する代替手段は、問題の処置を識別するための組織の問題追跡ツールを利用することである。前述の方法と同様、このツールは、どのリリースが問題を処置しているのかを識別できなければならないし、欠陥への処置と機能強化の区別、また製品のソフトウェア変更を要求する問題処置とそ

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.

うでないもの(例えば紙に書かれた文書の変更、第三者のソフトウェア変更)も同様に区別できなければならない。顧客が見つけた問題と内部で見つかった問題を追跡するために別々のツールが使われるなら、この 2 つのシステムからの計数値は、報告する SFQ の計数値を得るために合算する必要がある。

8.1.2.2 SFQ 測定値はどのように利用できるか?

ソフトウェア問題処置品質測定法は欠陥ありと決定されたソフトウェア問題処置の比率である。その比率が高くなるほど、問題を是正するためのソフトウェア問題処置のインストールによって、修正が失敗するか、追加の問題がネットワークに持ち込まれるリスクが高くなる。

SFQ 測定法を使うとき、特に目標を設定して継続的改善を推進しようとするとき、重要なことは、TL 9000 パフォーマンスデータレポートの平滑化ルールを考慮し、平滑化された平均値を使用することである。月単位の短期的データは、欠陥のある問題処置の発見の遅延のため、ソフトウェア問題処置品質の傾向を正確に表すにはあまりにも変動が多すぎることを示している。この変動は、年に数回だけの問題処置しかリリースしない製品においてはより顕著に表れる。

SFQ 測定値は下降傾向、究極的には 0 に近づくのが望ましい。この傾向が見られなかったら、組織はプロセスの改善の可能性を認識するために、欠陥のある処置について欠陥分析をすることを考慮することが望ましい。この欠陥分析は、実施した問題処置がなぜ機能しなかったのかに焦点を当てるべきで、最初の問題の原因が何かではない。詳細な欠陥分析では、問題の根本原因、欠陥のある処置が利用される前に検出できた筈の対策、及び欠陥のある問題処置がソフトウェアに導入されるのを予防できた筈の対策を識別することが望ましい。

欠陥のある問題処置の比率に突出部分があれば、組織は問題が特定のリリース、顧客、製品環境等に関連しているかどうかを分析する高レベルの欠陥分析を実施できるであろう。これは、個別の欠陥をより詳細に分析をするよりも、もっと低いコストで改善の余地の可能性を識別する助けになるであろう。

8.1.2.2 How does Product or Service in Retirement Affect SFQ (MHB 5.6)?

With Measurement Handbook R5.6, when a product or service is in Retirement Phase (or beyond), data will be excluded for this product or service from the monthly data submission. This exclusion also applies when a software product or a particular release of a software product is no longer deploying new features.

For SFQ, any defective fix associated with a release that is in Retirement Phase (or beyond) at the time the fix is found to be defective is excluded in the measurement numerator.

The SFQ measurement denominator includes the total number of software fixes that became available for general release during the 12 month period leading up to and including the report. Any fixes associated with products in Retirement Phase (or beyond) at the time the fixes were made available are excluded in the measurement denominator.

8.1.2.3 運用終了の製品又はサービスはどのように SFQ に影響するのか(MHB 5.6)?

測定法ハンドブック R5.6 によって、製品又はサービスが運用終了期間（又はそれ以降）の時、この製品又はサービスのデータは毎月のデータ提出から除外されることになった。この除外は、ソフトウェア又は特定のリリースがそれ以上の新機能を追加することが無くなった場合にも適用する。

SFQ 測定法の分子では、その時点で運用終了期間（又はそれ以降）にあるリリースに関連する欠陥のある修正は、修正に欠陥があることが判明した場合は、除外される。

SFQ 測定法の分母では、報告に至るまでの 12 か月間に一般リリースで利用可能になったソフトウェア修正の総数が含まれる。修正が利用可能になった時点での運用終了期間（またはそれ以降）の製品に関連する修正は、測定法の分母から除外される。