Comparing Reliability Levels of Software Releases*

Pete Rotella
Cisco Systems, Inc.
2700 Kit Creek Road
Research Triangle Park, NC, USA 27709
+1-919-392-3854
protella@cisco.com

Sunita Chulani
Cisco Systems, Inc.
125 West Tasman Drive
San Jose, CA, USA 95134
+1-408-424-6802
schulani@cisco.com

ABSTRACT

An intuitive method is needed to achieve buy-in from all sectors of Engineering for a way to gauge release-over-release change for a given product's sequence of releases. Also, customers need to know if there are extant releases that are more reliable than the ones they already rely on in their networks. A new Release-Over-Release (RoR) metric can both enable customers to clearly understand the reliability risk of migrating to other available releases, and also enable Engineering to understand if their software engineering efforts are actually improving release reliability.

CCS CONCEPTS

• **Software and its engineering → Software creation and management → Software post-development issues**

KEYWORDS

Software reliability; customer experience metrics; release selection; release management; customer-found defects; SWDPMH.

1. INTRODUCTION

Software development teams can benefit from using two similar, but clearly different, ways to gauge field reliability of a product's software release. A 'customer pain' metric is desirable to:

- Ascertain the reliability impact to the release's entire install base (IB) for a given product
- Factor in the disruption and other negative effects of bugs that influence many customers
- Help in establishing long-term reliability goals for the release.

A 'bug count' metric is also desirable. Its strengths lie in its ability to:

- Focus attention on specific customer-found defects (CFDs)
- Enable Engineering teams to reconcile and validate CFD counts.

At our company, we call the 'customer pain' metric 'SWDPMH' (i.e., software defects [encounters] per million usage hours per month). This is a normalized metric, with the numerator equal to the 12-month moving average of the number of CFD 'encounters.' 'Encounters' is the number of times all of the CFDs are experienced by all customers for the time period in question. For example, one CFD experienced by 10 customers adds 10 to the numerator of the calculation. (If the product's release is in the field for less than 12 months, we use all available months' data.)

The SWDPMH denominator is the number of millions of hours the product/release is used, by all customers, during the previous month. The combination of the numerator and denominator here gives a normalized feel for the (lack of) reliability 'pain' experienced by the entire customer install base of each product/release combination. See [1,2] for more information about the SWDPMH metric.

The 'bug count' metric is found to be useful by the engineering teams. The numerator of this metric is the number of CFDs experienced by all customers during the previous 12-month (or as many months as are available) period. The denominator is the previous month's install base for the product/release combination. The combination of the numerator and denominator here give a normalized feel for the reliability burden that can be specifically addressed by engineering.

Why do we need two similar metrics? Many teams do relate better to the CFD/IB metric, compared to the SWDPMH metric, since counting and tracking CFDs is already part of the normal release readiness process. This is a more intuitive metric than SWDPMH, and easier to explain and focus on.

But SWDPMH is also needed to properly gauge the 'total effect' of the CFDs on all customers, and the SWDPMH trends are useful in establishing, and tracking progress against, yearly goals, which are best determined by the overall fleet impact of the CFDs.

In addition, there is a third metric need: The need to compare releases – to enable 'release-over-release' (RoR) reliability comparisons. Since we have two commonly-used release reliability metrics, it makes sense to combine the two to have a single value to use to compare releases.

SWDPMH and CFD/IB, however, can have substantially different values. How do we combine two somewhat dissimilar numbers, which are not of the same scale, in a way that gives roughly equal weight to each? An effective way to

CESI'18, May 28, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5736-4/18/05…$15.00
https://doi.org/10.1145/3193965.3193968.

accomplish this is to take the geometric mean of the two components, which is the square root of the product of the two values. The geometric mean of SWDPMH and CFD/IB is a reasonable representation of the release's reliability for a given product. (The geometric mean is a way to find a reasonable combination of disparate numbers. As an example, the geometric mean of 2 and 8 equals 4, the square root of the product of 2 times 8.) See the APPENDIX for more information.

2. SWDPMH AND CFD/IB

Release-over-release (RoR) comparisons are essential, since we need to know:

- Which releases to recommend to customers
- Which releases have experienced either superior or problematic development and test practices and processes.

When we compare a fairly new release with older ones, we recalculate the RoR values for the older release by including the same number of months, in the calculation, in the field as the newest one. This enables a fair apples-to-apples comparison, since the natural exponential decay of both SWDPMH and CFD/IB curves over time follow a fairly predictable decay pattern.

Tracking SWDPMH, CFD/IB, and RoR values for each product/release combination enables us to satisfy the need of engineering to monitor both customer-found bug count and bug encounters, and also enables engineering and quality assurance management to set quality goals for individual releases and to recommend specific releases to customers.

Here is an important problem we are trying to solve: Customers are often highly risk-averse, tending to remain on older releases that do not contain important fixes and features. How do we convince customers that newer releases are reliable and present an acceptable risk?

After the software is released to the field, we need a way to ascertain if the new release is more reliable, has about the same reliability, or is less reliable than:

- The previous similar release – with roughly the same number of new features and release strategy, or
- The typical release produced over the past several release cycles, or
- The release(s) that customers would likely want to replace with the new one.

The two key metrics mentioned above are included in the release-over-release calculations:

- SWDPMH – our primary customer experience metric is a normalized 'customer pain' estimate that measures not the number of field bugs but the number of times field bugs are encountered
- CFD/IB – Customer-found defect count is the number of bugs found in the field normalized by install base.

CFD count is an unnormalized metric, so not as good an estimate of overall 'pain' as SWDPMH, but CFD count is considerably more meaningful to development and test engineers. Focusing on reducing CFD count is cleaner and easier to understand than reducing SWDPMH. This is a key point – we need to achieve buy-in from the engineering teams, and although SWDPMH correlates very well with customer sentiment (as measured by our yearly customer software satisfaction survey), we need a highly intuitive metric (i.e., a reliable release over release metric) to augment the more complex and less intuitive one (i.e., SWDPMH).

Focusing first on SWDPMH, here is an actual example of the type of reliability improvement progression we hope to see in a product's release sequence.

See Figure 1:

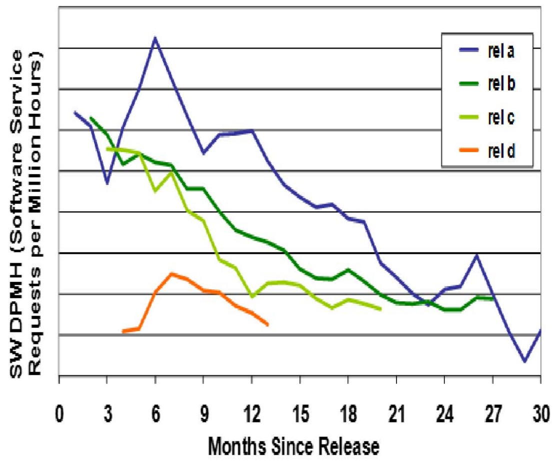

Figure 1: SWDPMH values for a product's release sequence

Including error bars, here is another example, with a desirable reduction of SWDPMH (~20% decrease over 12 months) over time in the field, with periodic slight setbacks/increases that are the result of published maintenance releases (that typically contain small sets of new features). See Figure 2:

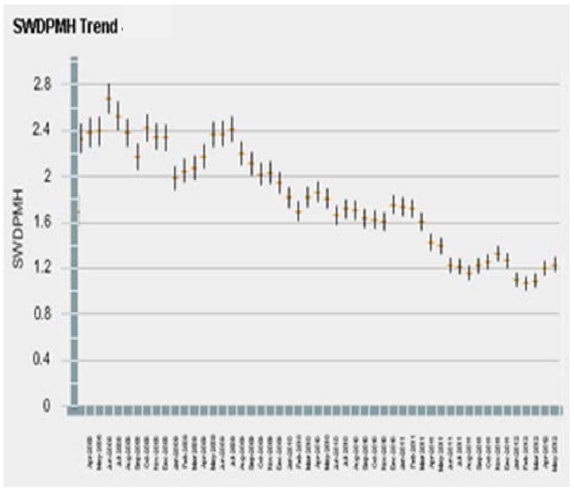

Figure 2: Steady improvement over time for SWDPMH

The following example shows a lack of improvement (~5% increase in SWDPMH over 12 months) for a different product's release, see Figure 3:

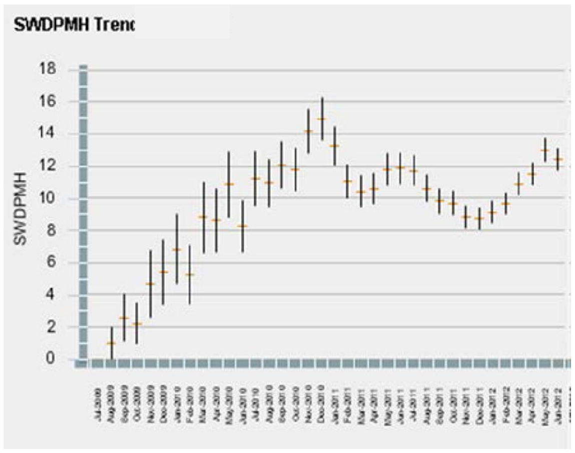

Figure 3: Lack of improvement over time for SWDPMH

3. RELEASE-OVER-RELEASE EXAMPLES

As mentioned above, since the SWDPMH and CFD/IB values can be quite different for a given product's release sequence, we take the geometric mean of the two values for a given release and call this the Release-Over-Release (RoR) metric. The aggregate metric is effective in characterizing release reliability, not only from an intuitive/actionability point of view, but also from its correlation with customer sentiment. Both SWDPMH and RoR correlate in the 70%-75% range with customer software satisfaction survey results.

The figure below (Figure 4) shows the SWDPMH value, the CFD/IB value, and the geometric mean aggregate value for a major router product. The aggregate value is our key metric for comparisons, and for this product's historical releases, we see a gradual decline in reliability (i.e., increasing SWDPMH and CFD/IB values) from releases 4.1 through 5.2, then a recent improvement in release 5.3.

The one standard deviation error bars for the RoR metric (in white color), when 'overlapping' between releases, represents a situation in which we cannot with sufficient certainty say that the releases have different levels of reliability. When the error bars do not 'overlap,' we can be reasonably sure of a release-over-release reliability delta. See Figure 4:

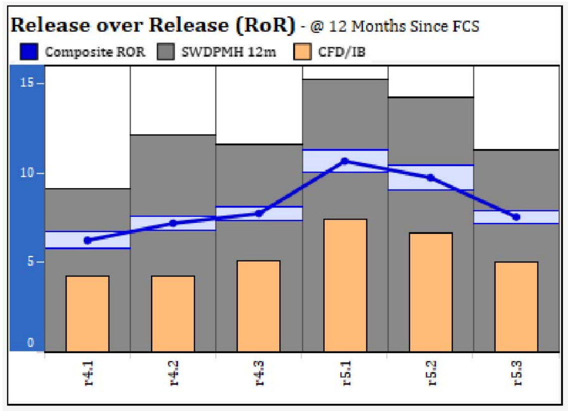

Figure 4: Example of reliability declining, then improving

The two figures below show increases in reliability for the release sequences for two different products, then show a slow-down in improvement. Overlap of error bars in some cases indicates uncertainty in true release-over-release improvement. See Figures 5 and 6:

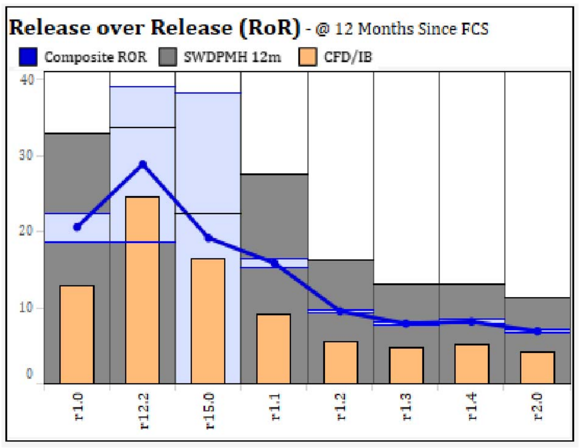

Figure 5: Reliability improvement, then uncertainty (1)

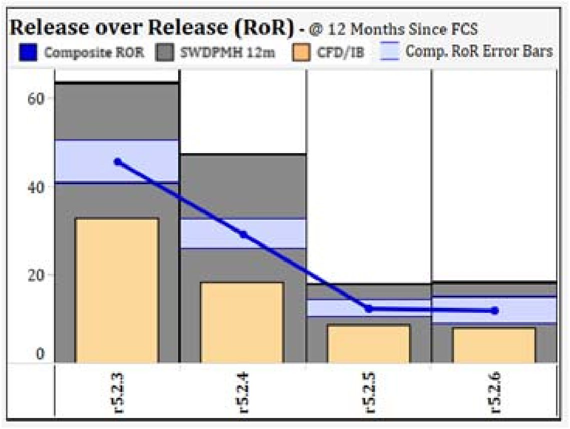

Figure 6: Reliability improvement, then uncertainty (2)

The two figures below (Figures 7 and 8) show gradual increases in reliability for release sequences for two different products, but sometimes show an overlap in the RoR error bars, therefore they demonstrate uncertainty in release-over-release change.

Figures 7 and 8:

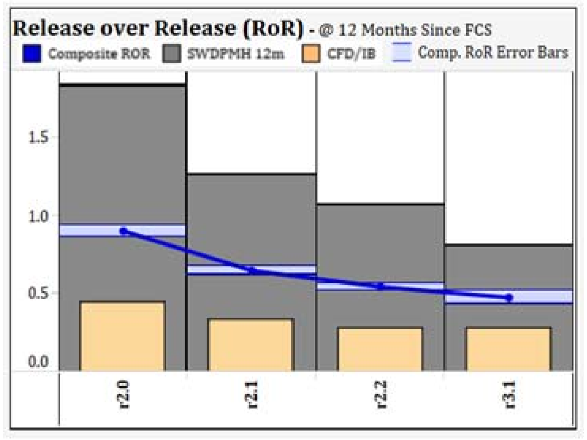

Figure 7: Example of reliability improving

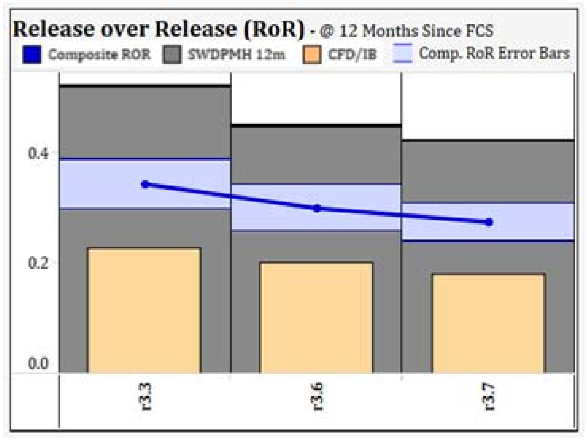

Figure 8: Example of uncertainty of reliability change

The two figures below (9 and 10) show gradual recent decreases in reliability, but some overlap in the error bars, therefore uncertainty in release-over-release improvement. See Figures 9 and 10:

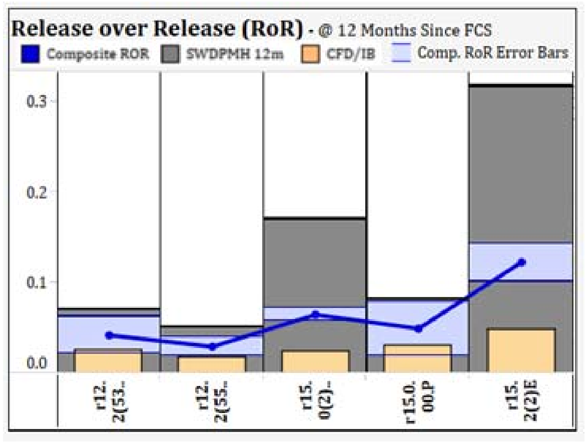

Figure 9: Reliability change uncertainty, then decline

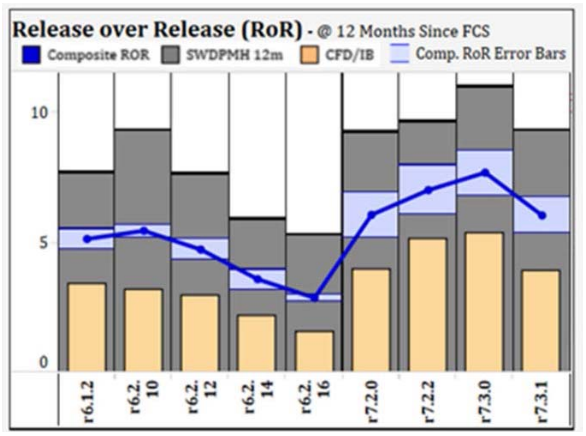

Figure 10: Reliability improves, declines, and uncertainty

4. SUMMARY

SWDPMH and CFD/IB are nonlinear and both the peak values and the exponential decline values are not easy to explain to engineering staff and customers. We need, essentially, a single scalar value for each of these key metrics, including error bars around these values. The Release-over-release (RoR) metric satisfies these needs. Also, we need a reference point, such as a 'software reliability class' value, to gauge progress toward best-in-class reliability [3].

APPENDIX

This is the rationale for using the geometric mean of the SWDPMH and CFD/IB values: For a likely case where we see a 30% increase in SWDPMH and a 50% increase in CFD/IB, for example, using the geometric mean, we would calculate a reliability decline of (1.3 x 1.5)0.5 = 1.4, a reasonable 40% worsening.

The real risk, however, is where a true improvement of, say, 30%, as measured by both SWDPMH or CFD/IB, would be magnified into a claim of 50% improvement. For example, if we have a SWDPMH of 1.0 and CFD/IB of 2.0, a 30% improvement in each would result in a SWDPMH of 0.7 and a CFD/IB value of 1.4. The mean value of these is 1.05, or an improvement of 47% (1.05/2). The geometric mean yields a value of 29% improvement (1.4/2.0), for the RoR metric. Using the geometric mean of the two metrics retains the desirable intuitive properties and keeps percent change claims bounded between the individual percentage changes measured by SWDPMH and CFD/IB.

REFERENCES

[1] P. Rotella and S. Chulani, “Predicting Release Quality," ISSRE 2014 (25th IEEE International Symposium on Software Reliability Engineering), Naples, Italy, November, 2014.

[2] P. Rotella and S. Pradhan, "Composite Release Values for Normalized Product-level Metrics," ISSRE 2015 (24th IEEE International Symposium on Software Reliability Engineering), Pasadena, California, USA, November, 2015.

[3] P. Rotella and S. Chulani, " Comparing and Goaling Releases Using Software Reliability Classes," QRS 2017 (16th IEEE International Conference on Software Quality, Reliability, and Security), Vienna, Austria, May, 2017.