Assessing End-User Reliability Prior To Product Ship

Mario R. Garzia, Mujtaba Khambatti, Mingtian Ni
Microsoft Corporation
Seattle, WA USA
{ MarioGar, MujtabaK, MiNi}@Microsoft.com

Abstract

Assessing the reliability of software products is critical in today's world. In this paper we describe an approach for assessing software reliability for mass market products such as the Windows® operating system. The approach described provides a good assessment of reliability in a short period of time by collecting data from a relatively small numbers of systems. The results of this approach have been used to make ship decisions for Windows Vista™ throughout the development cycle.

1. Computer Trends

Whether at work, home or on the road, people depend on software for their everyday life activities. This makes assuring that software is reliable an increasingly critical aspect of the decision to ship a software product. Veteran release managers will say that the decision to sign off on a product and release it to manufacturing has a lot to do with experience formed by acute observations of trends over the course of product development. However, especially of larger and more established products, it is important to formalize this decision based on key data points in the form of release / exit criteria that can be used by stakeholders to make an objective and well informed decision. There are several techniques that have been developed for assessing when software is sufficiently reliable to ship, most of these based on software testing with an appropriate operational profile [1]. For mass market products such as the Windows® operating system it is important to also consider an assessment based on pre-release software deployments that capture the large and varied set of scenarios in which it is used [2]. Doing such assessments is further complicated by the need to assess reliability in a relatively short period of time and with a manageable number of systems. In this paper we present an approach for arriving at clear, objective, and actionable reliability release criteria that have been successfully used for Windows Vista™.

2. Methodology

For the Windows Vista operating system we focus on end-users running desktop or laptop computers, for such users reliability means that the system runs whenever they need it. For them reliability is about disruption free operation, where a disruption is either a failure or a planned activity such as installing a software patch. While MTTF (Mean Time To Failure) would be a reasonable metric for this purpose if we generalize failure to include any disruption, it requires large runtimes and machine populations for the mean value to stabilize. These requirements typically cannot be met during bustling pre-release development cycles. As a result, the approach we use is to bucket user's computers in terms of the number of disruptions they experience per unit of time and then track the relative sizes of these buckets against set objectives. We do this by grouping computers into three groups, those experiencing excellent, good or poor reliability based on the number of disruptions they have experienced. The number of disruptions of disruptions that lead to a classification of excellent, good or poor in our case was based on expected improvement over our Windows XP SP2 baseline. We then measured computer disruptions for beta users and classified a user's machine into one of these groups based on the frequency of failures that the machine had on a particular build over a period of two weeks [3]. We are able to automatically detect these failures based on events Windows writes into each machine's event log.

3. Observations

The criteria were employed for every major Windows Vista milestone to assess reliability readiness and to identify reliability bugs. We developed an automated tracking and trending system that assessed reliability improvement towards a pre-defined baseline. This was important to release management, who were able to use the trends to develop an intrinsic feel for product reliability and make a final decision on

whether the product met its reliability objectives prior to each release.

We trended reliability improvement against a pre-defined target at every milestone of Windows Vista. Figure 1 shows the change in the percentage of computers that had an excellent/good/poor experience during the Windows Vista development process. The measurements are relative to expected improvements over the XP SP2 baseline. Over subsequent milestones, the percent of users with excellent experience increased above the baseline at the expense of users with good and poor experience. Another indicator of reliability improvement is that the percent of users with a poor experience dropped below the baseline at the RTM milestone.

We also studied the sensitivity of the release criteria metrics to changes in machine population and runtime. This allowed us to understand the minimum parameter values needed before the metrics would stabilize to accurate values. The graph in Figure 2 shows the variability when machine sample size was changed. We found that a minimum of 200-300 machines were required to provide good values (low variability). Such a user population is achievable during product development and allows for a quick way to accurately assess reliability pre-release.

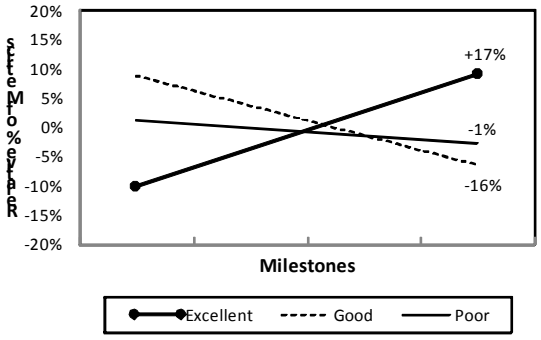

Figure 1: Trend lines showing reliability improvement across Windows Vista milestones relative to Windows XP SP2 baseline

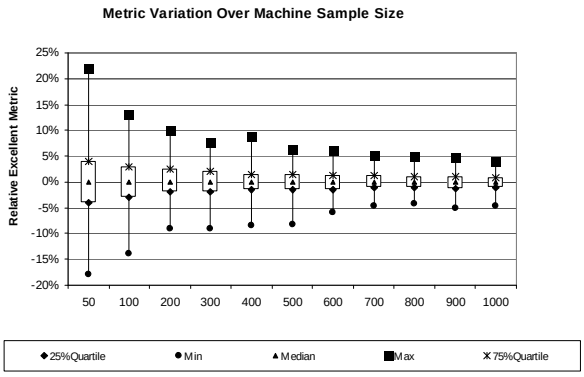

Figure 2: Metric variability as a function of sample size for the Excellent group

4. Conclusion

In this paper we discuss an approach to assess software reliability prior to shipping a product. This approach has the benefit of providing measures which are end-user focused (disruptions per unit of time) and can be assessed in a short amount of time and with a reasonable number of systems. The results from these studies were then used as the reliability ship criteria. These results were also compared with the results of post-release data coming from thousands of systems and with months of runtime, and found to be in agreement for comparable populations (similar user scenarios). Differences occurred only when new user scenarios were introduced in the deployed mix, highlighting the importance of having good user scenario coverage during the release criteria evaluation.

The approach presented in this paper was successfully used during the Windows Vista development cycle to assure that the product met reliability objectives prior to each major release.

5. References

[1] John D. Musa, "Software Reliability Engineering", Osborne/McGraw Hill, 1998.

[2] Brendan Murphy and Mario R. Garzia, "Software Reliability Engineering for Mass Market Products", DOD Software Tech News, 2004.

[3] Peter Galli, "Windows Vista Developer Talks About Quality",eWeek.com http://www.eweek.com/article2/0,1895,2055435,00.asp.