Parts of the SUM: a case study of usability benchmarking using the SUM Metric

Erin Bradner
User Research Manager
Autodesk Inc.
One Market St
San Francisco, CA
USA
erin.bradner@autodesk.com

Melissa Dawe
User Research Manager
Autodesk Inc.
One Market St
San Francisco, CA
USA
melissa.dawe@autodesk.com

Abstract

We present a case study of conducting a usability benchmark on a large, complex software application using a standardized, summated usability metric (SUM). We discuss nuances to the SUM equation and provide insight on key decisions including selecting tasks, setting target metric values, and translating z-scores into percentages when communicating results.

Keywords

Quantitative Methods, Case Study, Survey Design, Usability, SUM Metric, User Experience Design

Introduction: **using the SUM for benchmarking**

Through iterative usability testing, qualitative assessments of interface designs generate insights into how users' mental models are aligning, or failing to align, with the system model in the interface. Quantitative usability assessments commonly generate numerical tabulations and statistical analyses of the ANSI and ISO usability components: efficiency, effectiveness and satisfaction.

At [our company], we create large, complex software applications that are used by millions of [type of industry professionals] around the world. Most of the user research and usability conducted for these applications is focused on the newest features and latest user interfaces updates of each release (as is typical in the industry). From these studies, it is not possible to get an *overall perspective* of how the usability of our applications is increasing or decreasing over time. We began a usability benchmark program because we felt it was essential to measure the year-over-year change in usability of the core functionality of the software.

Usability benchmarking has benefited from relatively recent efforts of Sauro and Kindlund (2005a) to build a quantitative model of usability, which summarizes qualitative and quantitative components of usability into a single usability score, known as the standardized, summated usability metric (SUM). A SUM score derived from a given interface can be used as a benchmark against which subsequent designs are quickly and quantitatively compared.

At [our company], we chose to pilot the SUM metric for benchmarking because of the unique way the metric combines qualitative and quantitative measures of ease-of-use into one number. The formula appeared straightforward on the surface: compute z-scores for the measures we want to track (i.e. time on task, satisfaction), and then average those scores to get a single usability score. Finally, keep that score until the next release of our software, and re-run the benchmark and compare the scores.

While the SUM equation is well-documented by the authors, we found that using the equation with confidence required an understanding of nuances that were not obvious at the outset. For example, we were not prepared for the challenges of interpreting a z-score to a non-technical audience of business stakeholders when reporting the results of our study, and had to convert the final measure into a format our audience could readily understand. We also found ourselves lacking guidance around important decisions such as how to choose target levels for each measure. In this presentation we concisely describe solutions for the challenges of implementing the SUM metric that we encountered.

Background: deconstructing the SUM equation

Drawing an example from our research, the unit of analysis in a typical usability study of an engineer is to draw a floor plan and annotate the dimensions of each room.

* Net Promoter is a registered trademark of Satmetrix, Bain and Reichheld.

Data collected from users performing this drafting task might include the time users require to complete the task, a tally of errors they encountered while completing the task, users' overall success or failure in completing the task and users' responses to one or more post-task satisfaction scores. Consequently, the data includes both objective variables (time, errors, and completion rate) and subjective variables (satisfaction).

Combining these four measures of usability into one measure is non-trivial since it represents three different data types: *time* and *total errors* are continuous variables, *satisfaction* score on a Likert scale is an ordinal variable and *completion* rate, recorded as success or failure, is a binary variable.

Sauro and Kindlund (2005b) combine these four measures by first mathematically processing the measures so the variables of different data types (i.e. continuous and ordinal) can be combined. This process is known as standardizing the variables. Then, they equally weight (.25) and sum the standardized measures. This final sum is what they refer to as their single, standardized and summated usability metric, the SUM (Figure 1).

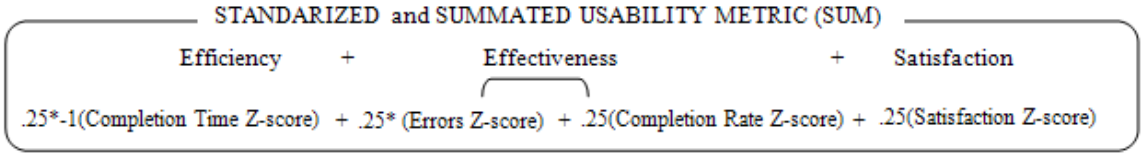

Figure 1. Weighted Quantitative Model of Usability Sauro and Kindlund (2005a)

For example, the z-score for time is:

$$\text{Z-score for Time} = \frac{\text{mean time} - \text{target time}}{\text{standard deviation}}$$

Figure 2. Z-score for Time

While the equation is straightforward, we encountered four nuances to using the equation that we document here.

Method, Participants & Setting

We conducted our first benchmark study in Spring 2008 with 20 participants. We selected 10 participants from two user profiles: experienced users, and beginning users or students. Our inclusion criteria for both user profiles was current and consistent use of our software application in their work or school projects. Our experienced users must have used our software for at least two years, and our beginning users/students must have completed at least one course in our software or used it in their work for a several months. Our software application is so complex and infused with domain expertise that it requires six months to one year to become proficient in use.

Our benchmark sessions were individual, in-person sessions conducted in our usability lab. Each session lasted 2.5 hours. Each participant was given a written workbook of tasks, and was asked to complete the tasks in the most efficient way they knew how. Participants were asked to voice when they were starting and stopping each task, but were *not* asked to think-aloud during the tasks (as this would have affected the task

timing). Participants were not interrupted during task execution unless they exceeded the cut-off time for that task. At the end of each task, the participant filled out a brief three-question satisfaction survey on the task. We had two dedicated observers for each session: one to record task completion times, and one to take notes on task success and user behavior (i.e. whether the participant's performance satisfied the task success criteria, and notes on non-failing errors and the participant's path through the interface).

Recommendations on Methodological Decisions

We continue this case study by briefly reviewing three methodological decisions we found ourselves faced with when conducting the benchmark and implementing the SUM metric. We quickly realized that these decisions would ultimately impact the utility and repeatability of the benchmark and thus required careful deliberation. We offer guidance based on our experience here.

Selecting benchmark tasks for your software

One of the defining properties of a usability benchmark is that it can be repeated year over year. This means that when you are designing a benchmark study, you must select tasks that can be repeated year after year. Without repeatable tasks, your data such as success rate and task time is incomparable, and you lose the benefit of the benchmark.

Creating tasks that don't change year after year is challenging for software that is constantly changing, gaining new features, and responding to evolving market needs. We developed specific criteria to guide the creation of benchmark tasks.

Benchmark tasks **should**:

- Examine the entire ownership experience (including install, migration , licensing, use, training, support)
- Emphasize areas of the product that you want to measure usability year after year (e.g. core functional areas)
- Be goal-oriented and high-level rather than specific step-by-step instructions
- Be granular enough (or use subtasks) to provide metric values on desired functional areas (e.g. you don't want to give users a single, monolithic two-hour task and end up with a single time-on-task number)
- When combined, give the user an end-to-end experience with the product

Benchmark tasks **should not**:

- Change release after release
- Call out a specific feature, since tasks should be relevant across many releases (the focus should be on functional areas instead of specific features)

For our benchmark tasks, we developed a written workbook that gave participants an end-goal for each task and minimal specific instruction on how to complete the task. In our UPA presentation we will provide examples of how we structured and presented our tasks.

Standardizing the scores

In order to make our findings digestible to a wide audience of business stakeholders, we converted our z-scores into percentages when we presented our results. This is discussed in more detail in Section V. Nevertheless, we were faced with another quandary: how should we set and communicate the threshold for "good usability"? We found setting the threshold to be a difficult undertaking. Using the z-score, an empirical mean that is equal to its target value (e.g. when the participants' task completion time was equal to the target time), results in a score of 50%. We were reluctant to have 50% as our grade for "good usability," primarily because it defied American cultural norms for a good grade and because we learned through peers in the field that others use values nearing 80% as their threshold for "good usability." To adopt the 80% threshold, we chose to adjust target values to be the "lowest acceptable" limit – a target of 80% success became 50% success. This reinterpretation of the target values and recalibration shifted our resulting SUM percentages up and enabled us to set the threshold near 80% for good usability.

Setting target values.

The discussion above underscores the importance of setting target values to the overall utility of the SUM benchmark. The target values we set were:

- Success rate target = 90% success for experts; 70% success for novices (although this target is no used in the SUM calculation since it is already in percentage form)
- Satisfaction target = 5 on a scale of 1 to 7
- Completion time target = the 80th percentile of completion time for users who rated the task with an average satisfaction score of 5 or higher. If 3 or fewer participants completed the task successfully, our completion time target was 75% of task cutoff time where the task cutoff time was the time limit to complete the task successfully; derived from 8 pilot sessions.

Deciding what to measure.

The SUM metric accommodates multiple components of usability, including but not limited to satisfaction, time on task, completion rate, and error rates. We chose to measure success rate, time on task, and completion rate. We eliminated error rates from our calculations for two reasons. First, calculating error rates demanded that we define errors explicitly at the outset of the study; we were unable to definitively define errors due to the complexity of the software and the nature of the knowledge work we were evaluating (engineering). Secondly, calculating z-scores required defining the opportunity

for errors. We were unwilling to set a target for errors due to the subjectivity involved in defining errors in the complex, multi-step tasks we were testing.

When deciding how to measure satisfaction scores, we elected to base our satisfaction z-score on a composite of three satisfaction measures each rated on a scale of 1-7. The measures were:

- Perceived performance: *How satisfied are you with the time it took to complete this task?*
- Ease-of-use: *How satisfied are you with the ease of this task?*
- User confidence: *How confident are you that you completed the task successfully?*

Four Critical Nuances to the SUM Equation

We encountered four nuances in the SUM equation that we needed to understand before we felt we could use the metric confidently. We address each nuance here.

Nuance 1: Inverting the Time on Task and Error Rate Scores.

One reason we were drawn to the SUM equation was its apparent simplicity. An inherent feature of the SUM metric is that all of the four components are calculated similarly: each is a z-score or percent of area under the normal curve.. Once we set out to apply the SUM calculation, we quickly learned that the time on task variable and error rates require special consideration.

First we considered time on task. When the mean time to complete tasks falls below the target time it indicates that the interface has exceeded expectations and is allowing users to complete the task faster than the expected. In this condition, as users complete a task faster than the target, the z-score for time grows increasingly negative. Mathematically speaking, when mean time to completion falls below the target time, the dividend in the z-score equation becomes negative, resulting in a negative z-score. Thus, negative z-score for task completion times signify better usability than positive z-scores. The same is not true for all of the other components of the SUM: when mean completion rate falls below the target for completion rate, usability is worsening; when mean satisfaction falls below the target satisfaction, usability is also worsening. Therefore, the sign of the time on task component of the SUM equation must be inverted to accurately reflect the ease-of-use (Figure 1).

Using a similar line of thinking, we next considered error rates. As error rates drop below the target, usability is improving. To account for this inversion, relative to the other components of the SUM equation, we must subtract the error rate from 1. Conceptually, what we are doing is calculating quality rate as opposed to error rate.

By inverting the time on task and error rate scores we achieved the necessary condition of continuity – as each z-score increases in value, the overall SUM will increase in value.

Nuance 2: Calculating or Looking Up

As we continued familiarizing ourselves with the SUM equation, we realized that two of the components of the SUM equation are based on calculated z-scores while others are looked up via the standard normal table. For example, to calculate the z-score for

time we subtract the target from the observed mean, divide by the standard deviation (Figure 2) and then we invert the sign.

Yet, for task completion rates we realized we can't calculate the z-score in the same manner. Rather Sauro and Kindlund (2005a) direct us to take the total number of successful completions and divide by total attempts then convert that number to a standardized value by looking up this percentage in a standardized z-table.

The reason for this look up step is that error rates and success rates are discrete variables, not a continuous variable like time on task. Therefore to calculate the z-score for success rate, the value is not a z-score but is a percent of area and the z-score is derived by looking up the percent of area in a standardized z-table (Figure 3):

$$\text{Value in standardized z-table corresponding to } \left(\frac{\text{successes}}{\text{attempts}}\right)$$

Figure 3: Completion Rate Equation

Nuance 3: The Enigmatic Sigma Shift

As we approached the end of our analysis, we realized that we needed to make a decision whether to use the post-hoc adjustment to the SUM metric called the sigma shift, as recommended in by Sauro and Kindlund. This sigma shift is a value that adjusts the SUM to account for a drift in the process over time. Typically this value is in the range of 1.5.

This value of the sigma shift in a formula like the SUM is most easily demonstrated if we consider a manufacturing process rather than a human process like working with software. In the manufacturing analogy, the SUM would not represent usability of software but would represent, for example, the quality of a finished product coming out of an assembly line. Over time, the equipment on an assembly line will experience wear and tear, producing lower quality products. The sigma shift moves the SUM value to account for this natural drift, thereby preventing us from over-estimating quality.

Returning to the equations for the SUM metric, the sigma shift could be applied to any component in the SUM metric. It is added to the z-score and for looks like this:

$$\text{Process Sigma} = \text{Z-score} + 1.5 \text{ sigma shift}$$

Figure 4: Sigma Shift

Ultimately, we elected not to apply the sigma shift. While we were aware of the longstanding tradition of using the sigma shift in manufacturing and other fields, we felt that we had not precedence on which to base the use of the sigma shift in usability. We considered user fatigue could be a corollary to machine wear, yet this could be countered by potential improvement over time in human performance through practice and skill development.

Furthermore, we considered that well-documented models of muscle memory in software use combined with our users' exceptional ability to combat drift, using keyboard shortcuts and dedicated controls on input devices, renders suspect the use of the sigma shift in our usability benchmarks.

Nuance 4: Reporting the Final Usability Measure

Once we had calculated our z-scores and run the SUM, we found ourselves in a quandary. Since the SUM equation is a sum of four z-scores, it is itself a z-score. Being a z-score, a SUM score of zero indicates the interface is meeting our target expectations for usability. A SUM of greater than zero indicates that the interface is exceeding expectations and a negative score indicates it is falling short of expectations. Furthermore, as a z-score, a value of 1 indicates that the score is one standard deviation above the mean, a value of 2 indicates it is two standard deviations above the mean and so on.

We developed several visual representations to communicate the z-score (Figure 5, for example). Still, our audiences were stumped. They lacked experience interpreting z-scores and thus were unable to interpret and apply the findings to their designs.

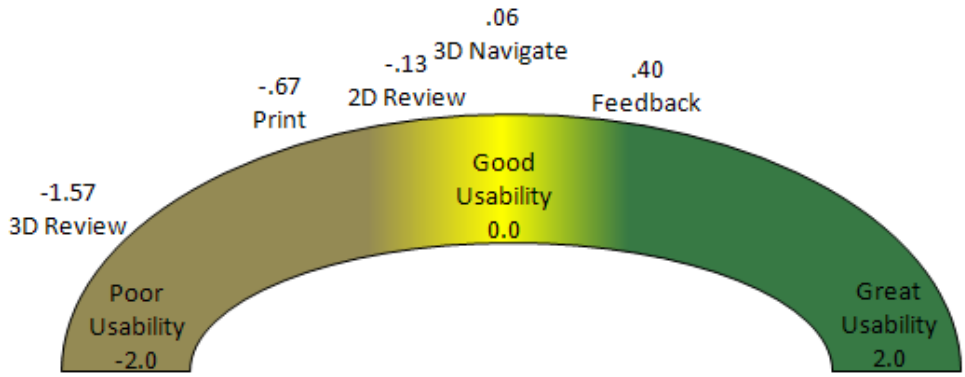

Figure 5: Sample Visualization of SUM Scores

It was at this point that we realized we had a choice. We could continue reporting our summative usability benchmark data as z-scores or we could convert to another format. We chose to convert the z-scores to percentages. We did this by converting each z-score to its standard normal cumulative distribution (note: we used the normsdist formula in Microsoft Excel®). Once converted into percentages we found our audience was better equipped to interpret the finding.

Conclusions

As expected, there were numerous other decisions that required research and debate as we piloted our benchmarking program using the SUM metric. Developing a usability benchmark for a large, complex software application is an involved process that requires time and a team of technical and business stakeholders.

In summary, we have reported some of our lessons learned conducting a usability benchmark on [our software app], and our experiences using the SUM equation. The nuances of SUM translate into the following four items to consider when using the SUM metric:

- Invert time on task and error rate scores before calculating the final SUM

- Recall that that error rates and success rates are discrete variables, not a continuous variable such as time on task. Therefore the z-score for these variables is a standard normal deviate and is derived by looking up the percent of area in a standardized z-table
- Apply a sigma shift if you believe that a natural drift, or tendency toward entropy, exists in the system you're measuring. If not, do not apply the sigma shift.

Consider your audience when determining how to report out your final SUM scores. Consider converting z-scores to percentages so your audience is better equipped to interpret the findings.

References

Sauro, J., & Kindlund, E. (2005a). A method to standardize usability metrics into a single score. In Proc CHI 2005, ACM Press (2005), 401-409.

Sauro, J. & Kindlund E. (2005b). Making Sense of Usability Metrics: Usability and Six Sigma. In Proceedings of the Usability Professionals Association (UPA 2005) Conference Montreal, Canada

Additionally, our research has drawn on various publications as background information, including:

Ebling, M. R. and John, B. E. (2000). On the contributions of different empirical data in usability testing. In Proceedings of the 3rd Conference on Designing interactive Systems: Processes, Practices, Methods, and Techniques (New York City, New York, United States, August 17 - 19, 2000).

Martin, R. and Weiss, S. (2006). Usability benchmarking case study: media downloads via mobile phones in the US. In Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services (Helsinki, Finland, September 12 - 15, 2006).

Sauro, J. (2006) Quantifying usability. interactions 13, 6 (Nov. 2006), 20-21.