What Is Performance?
Part 2: Qualities and Quantities

Tom Wilson

1 Introduction

This paper is an overview of the many aspects of system performance. A system can be almost anything (e.g., a computer, a car, an airplane, or a person), but the focus will be on systems that are or contain computers. *Performance* is a quality that expresses how well a system carries out its function. How well the system serves the user is one way to view performance. This is often the one that the users and/or customer takes regardless of requirements.

In Part 1 ([Wil11]), I discussed numerous perspectives of performance. Part 2 investigates qualities and quantities. A system has many qualities: color, size, shape, weight, availability, and affordability. Some of those are unchanging while others change with time or as a result of other factors. I will list many names of qualities that can be viewed as performance and group them for some comparison. Then I will discuss the importance of quantities to associate with qualities. Finally, I will introduce the concept of trading, which is essential in order to improve some qualities.

2 Qualities

A *quality* is a distinctive characteristic or attribute (e.g., color, size, shape). It can also reflect character or nature (e.g., simple, fast, available). The problem with qualities is that they give you some idea of the attribute, but they don't tell you much more. Subjectivity and the lack of precise definitions and standards are key contributors. Before addressing this problem, let's look at some qualities with the expectation that they might be vague.

Systems engineering refers to certain requirements as non-functional. They are colloquially called "-ilities" since many terms have this ending. A popular list appears in [wik, "Ilities", 2011]. The potential for qualities seems endless as we skim the list (it is getting longer every year). The silliness of many of these is illustrated by a comical quote:

> "Wrong! Wrong! Absolutely brimming over with wrongability!"
>
> – Arnold Rimmer in the British sci-fi television comedy *Red Dwarf*
>
> .

"Wrongability"!? And surely, you've heard of "drinkability"! Such terms emphasize the difficulty in expressing quality; terms are made up that just further the problem rather than solve it. I won't be solving the definitions problem here; I'll only be highlighting it.

[LDM98] says that the performance of a computer system can be described via 5 simple concepts: workload, response time, throughput, resource utilization, and resource service time. I think these are major aspects, but, by dealing with only these, they limit the perspective. However, this reference also says that a number of other qualities (specifically scalability, response time, throughput, access time, utilization, capacity, cost) all contribute to overall performance. It also gives some attention to different perspectives (e.g., individual user vs. groups) as well as the psychology of the perception of performance. This reference is better than most at highlighting the generality of performance.

I will divide qualities into *system* and *development* categories. However, this division may not be satisfactory. Qualities can easily be in both categories. For example, affordability can be applied to the system while it is in use (i.e., operations/maintenance costs are low) as well as to the development of the system. Within each category I might make subcategories and provide some examples. In many cases, some non-performance qualities may be thrown in for contrast.

2.1 System Qualities

System qualities are qualities that are applied to the system[1] itself. Such qualities often reflect the users' perspectives. They deal with the aspects of execution or composition (i.e., physical properties). As we stated earlier, performance is a quality that expresses how well a system carries out its function. Saying that a system is fast is focusing on its performance quality of responsiveness (or, is it throughput. . . ?). Saying that a system is colored blue does not address a performance aspect. Saying that a system is heavy does not address a performance aspect of the system, but might impact the performance of the user if he carries the system (i.e., laptop). This is where perspectives can muddy the water. I am not interested in drawing boundaries or standardizing terms here.

So what are some system qualities? Table 1 lists many, assigning them to four common groups. Definitions will not be given for any quality. In some cases, a quality appears in more than one group.

Table 1: Fundamental Performance Qualities

Time	Space	Computation	Dependability
Efficiency	Availability	Accuracy	Availability
Latency	Capacity	Precision	Maintainability
Promptness	Efficiency	Repeatability	Recoverability
Responsiveness	Scalability	Reproducability	Reliability
Timeliness	Throughput	Validity	Stability

So, let me nitpick the list a little. *Time* and *space* are fairly well known groups. What about *computation*? This is fundamentally about what the system produces or outputs. The *dependability* group concerns the health of the system (i.e., does it work?). *Availability* is in the *space* list when there is a capacity limitation that influences processing without compromising the health of the system. Such a system might output a message telling the user to submit an input at a later time. For this input, the system is not available, but for others it is.

You might ask if *throughput* is associated with *time*. By most definitions, the answer is *yes*. Typically, throughput is a rate—most likely, space-per-time. The numerator of the rate might come from an atypical measurement scale (e.g., users, transactions), but it probably has a space component in the system. Now, the discussion is starting to transition us into quantities, but let's not go there yet.

Let's look at some more qualities. Table 2 lists several other qualities associated with the system. Some deal with the operations and/or maintenance of the system, while others deal with physical attributes of the system.

Table 2: Other Qualities

Accessability	Administrability	Auditability	Durability
Degradability	Lethality	Learnability	Safety
Security	Size	Survivability	Sustainability
Transportability	Usefulness	Utility	Weight

In most cases, these are not performance qualities. Depending on your perspective, you might argue that they factor into the system's "effectiveness" (hey, that's not in any list!).

[Hus00] discusses the term *quality* in a vague way, yet it is similar to our definition of *performance*. This quality is an expression of "how good it is". The expression "quality of service" follows: How good is the service provided?

2.2 Development Qualities

Development qualities are qualities that are applied to the development of the system. There are fewer people perspectives that focus on these, but they get a lot of attention because of affordability. Table 3 lists many development qualities. Some of these qualities could be system qualities as well. It all comes down to actual definitions. I believe the lengthy list in [wik, "Ilities", 2011] exists because people are trying to split hairs with the definitions: "The *absurdability* term in the list is not what I mean, so I'll create the *stupidosidicity*[2] term instead."

[1] The system could be a service instead.

[2] Not to be confused with *stupidisidocity*.

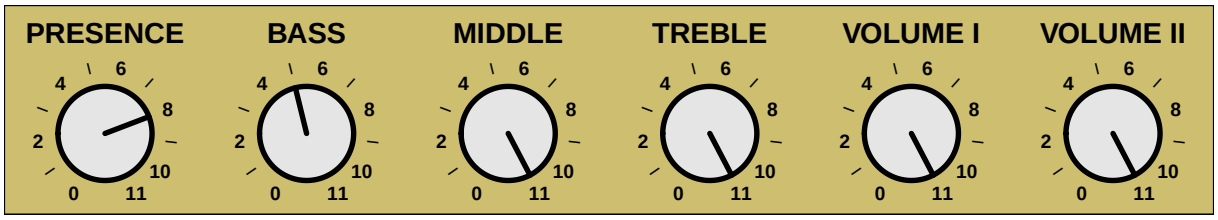
Table 3: Development Qualities

Adaptability	Affordability	Compatibility	Configurability
Customizability	Demonstrability	Deployability	Extensibility
Flexibility	Installability	Interchangability	Interoperability
Maintainability	Manageability	Modularity	Portability
Reusability	Scalability	Serviceability	Supportability
Tailorability	Testability	Understandability	Verifiability

3 Quantities

Qualities communicate some information, but not enough. Quantities are needed to tell us more. But we can still fall short if we are not careful. [wik, "Level of Measurement", 2011] defines four types of measurement scales: nominal/categorical, ordinal, interval, and ratio. Nominal/categorical scales are just names or categories with no relationship among them. Colors (e.g., green, blue, red, orange, etc.) are an example. Ordinal scales provide ordering, but only comparisons can be performed. A silly example is: slowest, slower, slow, fast, faster, fastest. Diamonds have a clarity grading scale: FL, IF, VVS_1, VVS_2, VS_1, VS_2, SI_1, SI_2, I_1, I_2, and I_3. Interval scales allow addition and subtraction, but not multiplication and division. Interval scales are relative with no absolute zero for a reference. Ratio scales are similar to interval scales, but have an absolute zero and support multiplication and division. Most of our familiar scales are ratio: time, distance, and weight.

Before we get more specific, let's improve the "humorability" of Part 2 by looking at a silly example of how numbers on a scale are not necessarily informative enough. If you have seen the movie "This Is Spinal Tap", you will remember this funny banter.[3] In the scene, Nigel is explaining the volume range of his amplifier to Marty:



> Nigel: . . . but it's very, very special because, if you can see, the numbers all go to 11. Look, right across the board, 11, 11, 11, and—
>
> Marty: —Oh, oh, I see. And most of the amps go up to 10?
>
> Nigel: Exactly.
>
> Marty: Does that mean it's louder? Is it any louder?
>
> Nigel: Well, it's one louder, isn't it? It's not 10. You see, most–most blokes, you know, will be playing at 10. You're on 10 here—all the way up, all the way up, all the way up—you're on 10 on your guitar. Where can you go from there? Where?
>
> Marty: I don't know—
>
> Nigel: —Nowhere. Exactly. What we do is, if we need that extra push over the cliff, you know what we do?
>
> Marty: Put it up to 11.
>
> Nigel: 11. Exactly. One louder.
>
> Marty: Why don't you just make 10 louder and make 10 be the top number and make that a little louder?
>
> Nigel: [long pause while looking at the amp] These go to 11.

A *quantity* is a specified or measured amount. When measuring a quantity, we will almost always want to use ratio scales. But let's look at measurements a little further. What we loosely call a "measurement" can really take on three forms: (1) a measurement on a ratio scale, (2) a rate, and (3) a proportion. A measurement on a ratio scale has a unit of measure (e.g., seconds, inches, pounds). That unit of measure could be the result of multiplication: Area can be expressed in square inches. A rate is the ratio of two measurements with different units (e.g., inches/second). A

[3] You might find the scene on www.youtube.com by searching for "Eleven Is One Louder - Spinal Tap".

proportion is the ratio of two measurements of the same units and results in no units (e.g., a high-definition television has 16:9 aspect ratio). The interesting thing about these different types of measurement is that each uses a different formula for computing averages.[4]

What about counts? Counts are effectively rates. The measurement in the numerator is a discrete (i.e., integer) measurement of some arbitrary unit (e.g., bit, person, transaction). The measurement in the denominator is a typical ratio scale measurement. Any time a counter is sampled in a computing system, it is really a count over an interval (usually time).

We will find that qualities can be expressed by all three kinds of quantities. Response time and capacity are measurements on ratio scales. Throughput and bandwidth are rates. Efficiency and availability are proportions (not only that, but these scales are bounded).

Quantities are the only way we can be (more) sure that we are talking about the same quality. Engineering requirements require quantities so that they are specific, measurable, and testable. Quantifying performance is necessary, risky, and often complicated. Whether or not a quantity is too high or too low is a different issue that I will not discuss.

3.1 Specifying Quantities

Quantities shows up in specifications, and perspective and terminology are important. The following terms are common:

- *Service Level Agreement* (SLA) - a legal agreement between a customer and a service provider. Quantities will be mentioned in the SLA and, hopefully, what is being measured is detailed.
- *Key Performance Indicator* (KPI) - industry jargon for a quality deemed important to a higher objective (e.g., success). Quantities are invariably associated with the named KPI.
- *Quality of Service* (QoS) - while this began as a network phrase, it has crept into other domains. Usually, the service has quality levels that can be compared.
- *Key Performance Parameter* (KPP) - a government term to indicate that an aspect of a system is important to the holistic view of the system. The Joint Strike Fighter F-35 has a KPP that indicates that the number of sorties achieved by the aircraft is critical to the success of the program.
- *Measure of Effectiveness* (MOE), *Measure of Performance* (MOP), and *Measure of Suitability* (MOS) - these are all government terms that attempt to quantify various system qualities.
- *Technical Performance Measure/Metric* (TPM) - an engineering term occurring in government work. It is intended as a project management tool to track how well a design aspect meets a design goal. For example, there are often TPMs for KPPs to help management understand if the performance goals are going to be met. TPMs are an improper means to specify requirements. TPMs may contain estimates (often the result of modeling) in early design stages rather than measurements. Often, target and threshold quantities occur in the TPM specification.
- *Performance Based Logistics* (PBL) - a government business strategy for improving total cost of ownership. This strategy attempts to make development consider operations and maintenance costs (especially logistics) during the design. Performance quantities are bound to show up in a specific PBL contract.

The completeness and/or thoroughness of the list is not important. Understanding perspective is. In most of these situations, someone is trying to express performance aspects of a system. Gaps in the definitions and specifications are likely to exist, leading to misunderstanding. As I said before, quantifying performance is necessary, risky, and often complicated.

3.2 Metrics

The word *metric* may be used when referring to a quantity. A metric should be a measurement, but sometimes it is a statistic. Older literature may use the phrase "performance index" rather than "performance metric".

Performance metrics may be categorized as *internal* or *external*. Internal metrics may have limited value if they do not link to external metrics or to an understanding of the workload (possibly making them surrogate measures). For example, if a CPU is highly utilized, this is considered to be a good use of the resource. However, if the load is light, then the CPU is likely to be saturated when more load exists. It is best to assess internal metrics with supporting metrics and additional

[4] I won't go into that here, but the different averages are: arithmetic, harmonic, and geometric.

information. External metrics can be worthless if they are not well-defined. "Concurrent users" is a good example because the definition of "concurrent" is inconsistent.

[Mol09] divides a few metrics into *service-oriented* and *efficiency-oriented* indicators. This is essentially the same as external and internal.

4 Quality Trading

"Fast, good, cheap: pick two" is commonly quoted to illustrate that you cannot have everything. However, things are not that simple. Cost is almost always a factor, but sometimes a lot of money spent can return a lot of value. Nonetheless, I want to look at the concept of quality trading. Qualities often compete with each other. In order to improve one, you have to sacrifice another. The most common is the time-space trade-off of algorithms and data structures.

Figure 1 conceptually illustrates some relationships. One example illustrating the competition concept is error detection/correction for a network. The approach increases space (i.e., more bits) to improve the lack of dependability of the network (e.g., transmission errors). Because there are more bits, more time is required to transmit the messages. The approach allows some errors to be detected and corrected. This increases the accuracy of the network and can save time by correcting the errors rather than retransmitting the message. This is the common example of where some individual response times can be increased, but overall throughput is improved.

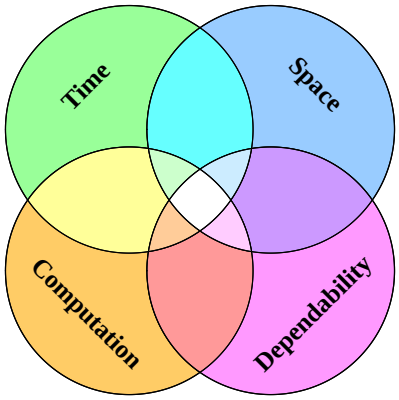


Figure 1: This two-dimensional figure illustrates the dependencies among competing qualities. A better illustration would use a dimension for each quality.

Many non-performance qualities have a performance impact. Security and safety tend to add function, which increases work and, therefore, decreases performance (e.g., increased response time). However, without these qualities, the system may be "down"—also decreasing performance.

This is not right versus wrong. It is a compromise toward a higher objective: the effectiveness of the system. Often, qualities cannot be quantified, so the trade is subjective. We call this *art* (i.e., experience), but it is risky art.

Some objective drives how to balance the qualities. We are not concerned with how to balance the qualities here, but it is effectively achieved by a trade study. Figure 2 illustrates that the system effectiveness is a balancing of the other qualities. The relative sizes of the qualities indicates their supposed importance.

[Kin03] shows a figure (attributed to someone else) combining Figure 1 and Figure 2 where three qualities (usability, utility, and likability) are balanced against cost. The intersection of the 3 qualities is labeled "speed" (evidently, not a quality on its own). The example highlights the same thing: qualities compete and they must be evaluated (or balanced) against a higher objective.

Sometimes qualities can be expressed by a common measurement (often cost). In that case, the trade can be done by means of a table or a chart. This approach is more objective if the expression of each quality is accurate. Figure 3 shows a notional trade-off curve. In this situation, qualities A and B have curves associated with them as they are evaluated in some way. The combination of the two results in the third curve. Here, minimizing the value on the third curve is the objective.

Trading two qualities is not always this simple. Introducing more qualities is usually intractable. A trade study attempts to represent many qualities with a single number. The result tends to be very subjective. Refer to [wik, "Trade study", 2011] for a brief introduction to trade studies.

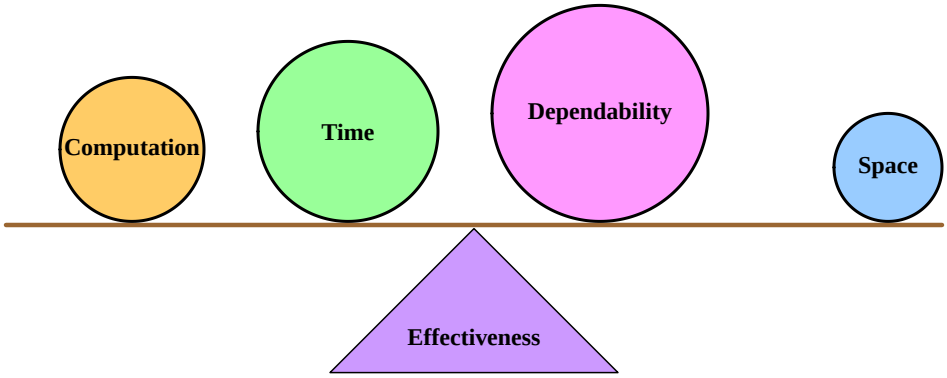


Figure 2: This figure illustrates the concept of balancing foundational qualities in order to optimize a complex quality. Unfortunately, the foundational qualities (i.e., the circles) are probably not independent (as illustrated in Figure 1).

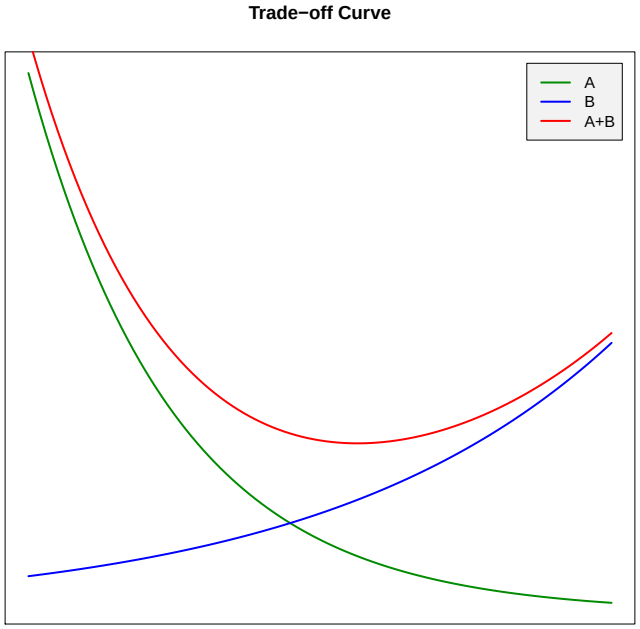


Figure 3: A notional trade-off curve. A and B are competing qualities expressed on some measurement scale. The objective is to minimize their sum. A possible example is: A=response time, B=throughput, and the measurement scale reflects cost.

When performance is a priority (or becomes one due to it being excessively poor), (1) inefficiencies must be removed, (2) other qualities must be compromised, and/or (3) additional resources are required. The final solution is to redesign the system. Some common trading examples:

- Page size in a virtual memory system
- Block size on a disk
- Number of error detection/correction bits for a network
- Dynamic memory garbage collection
- Number of processors in a multiprocessor
- Object-oriented programming vs. procedural programming
- Replicating data vs. sharing data
- Image quality vs. image size for web pages
- Compressing and decompressing transmitted data
- Encrypting and decrypting data

What is being traded may not be obvious. In some cases, there are many subjective pros and cons.

5 Conclusions

In Part 1, I discussed various performance perspectives with the hope of opening our eyes to the fact that we are not all on the same page. In this part, I listed many performance and non-performance qualities to help us better understand just how general the term *performance* is. Certainly, we want to focus on specific qualities in our conversations. Then I highlighted how integral quantities are to the qualities. It is easy to think we have quantified something only to learn that (1) the quantity is still vague, or (2) it wasn't the right quality to begin with.

To better understand qualities and their associated quantities, numerous metrics may be necessary. As in Part 1, being on the same page is important when discussing metrics since they can be deceiving. Finally, I talked about quality trading and gave some examples.

A couple of reports were very helpful in helping me understand qualities. They go far beyond this paper's topic. [Fir03] gives many quality definitions, but definitions are always subject to dispute (depending on one's perspective). [Eid05] discusses some existing non-functional (requirements) taxonomies, one of which is [Fir03].

Bibliography

[Eid05] Petter L. H. Eide. "Quantification and Traceability of Requirements". Technical Report TDT4735, Norwegian University of Science and Technology, 2005. `http://www.idi.ntnu.no/grupper/su/fordypningsprosjekt-2005/eide-fordyp05.pdf`.

[Fir03] Donald G. Firesmith. "Common Concepts Underlying Safety, Security, and Survivability Engineering". Technical Report CMU/SEI-2003-TN-033, Carnegie Mellon University, December 2003. `http://www.sei.cmu.edu/reports/03tn033.pdf`.

[Hus00] Geoff Huston. *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks*. John Wiley & Sons, 2000.

[Kin03] Andrew B. King. *Speed Up Your Site: Web Site Optimization*. New Riders Press, 2003.

[LDM98] Chris Loosley, Frank Douglas, and Alex Mimo. *High-Performance Client/Server*. John Wiley & Sons, 1998.

[Mol09] Ian Molyneaux. *The Art of Application Performance Testing*. O'Reilly Media, 2009.

[wik] Wikipedia. `http://en.wikipedia.org`. Specific page and last date referenced are noted in each citation. *Caveat lector: Because contributions to Wikipedia are not necessarily peer reviewed, readers should always validate what they are reading by other established sources. The references are given because of their simple accessibility.

[Wil11] Tom Wilson. "What Is Performance? Part 1: Perspectives". *CMG MeasureIT*, Issue 8, 2011.

About the Author

I am a performance scientist at Lockheed Martin. I have spent 10 years in software engineering and the last 8 years in systems engineering. I have worked with performance critical systems for soft-realtime visual simulation, biometrics processing, and military logistics and maintenance transaction processing. I have written many papers for CMG's MeasureIT, quarterly journal, and annual conference (they're on the site below). I tend to focus on scientific principles and general techniques applicable to all technologies.

This series is intended to be the introductory part to a free performance e-book called *Performorama*. The book is only in the planning and drafting stage[5] at this time and can be found at `www.performorama.org`. I plan to stop writing for MeasureIT for a while, but will take some requests for topics for 2012. If your request falls in line with the material planned for the book, you will improve your chances of a response. I can be reached via e-mail at dr_ziggy@att.net. I am also on www.linkedin.com.

[5] Set your expectations to "low"... uh, oh, that's not a very definitive quantity!