



**A reader asks:** We've followed your advice and created a staffing schedule using the "Erlang C" tables. It works great some days, but terrible on others. We were under the impression that this was a very accurate way to schedule our support center. Why is it so inconsistent?

**Dave's answer:** I'll answer that in two parts. First, I need to explain some of the challenges that come with using Erlang or any staffing formula in a support center environment. These challenges contribute to the fluctuations you're seeing and, to a degree, are unavoidable. Second, I'll explain what to look at to determine exactly what is causing you to miss your service-level targets.

There's an inherent weakness in Erlang C and other staffing formulas. They are all based on "probability theory." They use probability theory to predict how many calls will arrive at certain times. When dealing with high volumes, that works just fine. However, when you're dealing with lower volumes, the accuracy of the predictions is diminished. The most difficult environments to predict are those with either low volume or long handle times. Unfortunately, that describes most technical support environments. In fact, they often suffer from both maladies—low volume *and* long handle times.

I'll explain the issue using another, more familiar, application of probability theory. If you were to roll two dice one time, the chance of hitting a seven is fairly low. If you roll three times, the odds are better, but still there's no assurance you'll hit the seven. How-

ever, if you roll a dozen times or a hundred times, the odds are very good that you'll hit that seven. In fact, if we tracked all the combinations in one hundred dice rolls, the odds are very good that there would be a fairly even distribution. At a thousand rolls, the odds are even better.

The first step in staffing is predicting how many calls will be received in a specific period (usually an hour) and when those calls will arrive. As with the dice roll, the more volume we expect, the more likely the distribution will be even. If you were to receive 60 calls in an hour, they probably would not arrive perfectly equally...one every minute. However, you probably would not have any big gaps (no calls arriving for a long time) or big "clumps" (many calls in a short span of time). The odds are good that they would be spread out fairly equally throughout the hour...good, even distribution. In a large call center that receives hundreds of calls per hour, the distribution would be very even and/or predictable.

However, let's say you were only to receive four calls in one hour, and those calls take an average of 15 minutes to handle. If we could be certain that they will arrive evenly (every 15 minutes), then we could handle the workload and provide fast response (short hold times) with only one agent. However, the chances of the calls arriving every 15 minutes are very slim. It is more likely that you would see long periods, like 20 to 30 minutes, without a call. Then, the odds are just as good that you'd receive two or three calls within a five- to 10-minute period.

Erlang "knows" this, estimates the probability of when those calls will arrive, and then tells you how many people need to be staffing the phones in order to achieve your desired service level. If you refer to the Erlang C Tables (e-mail me if you can't find a copy) to determine how to handle those four 15-minute calls, it says you need four people staffing the phones to avoid the hold times from going over five minutes. Four people to handle four 15-minute calls! That would result in 25 percent utilization—one hour of work spread across four people. Why would Erlang tell you that? Because it has calculated the odds of all four calls overlapping, requiring all four agents to avoid a five-minute or greater hold time. You would be overstaffing the phones to ensure that you meet your service-level targets. That also means you'd have the ability to absorb some variances. If you actually received five or six calls, and they arrived in a reasonably even distribution, your staff could handle that spike.

That's a fairly dramatic example. Let's bump up the volume a bit and assume that you typically receive 20 calls during a one-hour period. Erlang would assume a more equal distribution, and it would calculate a reduced probability of big clumps. Therefore, it would not overstaff to the same degree as the previous example. That means you don't have as much cushion...not as much tolerance for variations. If that's the case, a few extra calls could cause you to miss your service-level target. Or, if the calls did arrive in a clump, you could have the same issue. Erlang

determines the staffing for the probable...not the possible.

Okay, now let's talk about how to trace the cause of missing the service-level target (SLT). There are three components involved in achieving the SLT—the workload, the arrival pattern, and the staffing. If all three happen as planned, then Erlang C has proven to be extremely accurate. If you didn't achieve the expected SLT, then one of those three components varied from the forecast.

I usually look first at workload. Not only is it very likely that it varied from the expectation, it is also one of the easiest to check. Workload is actually the result of two factors—call volume and handle times. So let's say the forecast was to receive 10 calls and the average handle time was predicted to be 15 minutes. That's 150 minutes of workload ( $10 \times 15 = 150$ ). If the actual calls received were 12, that's a 20 percent increase in workload ( $2 \times 15 = 30 / 150 = 20\%$ ). That's enough to have a noticeable impact on your service levels. Or, if the actual handle time was 18 minutes, you'd see the same effect. So, in reconciling your results, the first thing you should look at is the actual workload. Was it what you expected? If not, rerun your staffing calculation using the actual workload and see if the predicted service level matches your actual results.

The second possibility is that actual staffing was different from the planned or required staffing. This happens more often than most managers realize. In most cases, the cause is innocent and unintentional. Maybe you were supposed to have five engineers on the phones. But what if someone

went to the restroom, then stopped to get a cup of coffee on his way back? They were only gone for 15 minutes, but you were short-staffed during those 15 minutes, and that's sometimes enough to cause the calls to start backing up. Another common scenario is that one engineer got stuck on a really tough issue and was on the phone with one customer for 30 minutes or even the full hour. That means you were essentially understaffed for that period. The remaining calls had to be handled by four engineers...one less than Erlang assumed. That's enough to have a serious impact on the service levels. Now, put the two together, which is one engineer was on a long call and another took the restroom/coffee break. That could have devastating consequences.

The third possibility is that the call arrival pattern varied dramatically from the expectations. While the Erlang-recommended staffing level usually can handle normal variations, an extreme could cause a problem. For instance, if you expected 20 calls in an hour, but then received 10 of those calls in a 15-minute period, that would probably cause you to miss your SLT. Now, please note that I recommend looking at this last. That's because it is difficult to monitor or verify, and it is impossible to control.

So, if you've used Erlang to determine your required staffing, and then you don't achieve the expected service levels, look to those three factors to determine the cause. Once you understand what's causing the problem, you may be able to avoid it in the future. You can't avoid call spikes or arrival pattern variations. However, you can

be prepared for them. You also can educate your staff and have them work together with you to avoid some of the common causes of short staffing.

All that said, I don't want to give the impression that I'm negative about Erlang. In spite of the flaws, it is still one of the best tools available for determining appropriate staffing levels, particularly for smaller organizations. There are some new, more sophisticated software tools available. However, they are out of reach (budget) for many smaller support centers (see my previous column [January/February 2003] about justifying workforce management software). For the small center, Erlang is still your best bet—just be aware of the potential problems. ♦

**Have tough question?** Submit your question to Dave by e-mail at [dave.brown@SupportCenterU.com](mailto:dave.brown@SupportCenterU.com). Dave will respond to all inquiries, and if your question is selected for publication, you'll receive a complimentary copy of his book, *Optimizing Support Center Staffing*.

---

*Dave Brown is a management consultant, teacher, and writer. He teaches management training programs for Support Center University ([www.SupportCenterU.com](http://www.SupportCenterU.com)). He also consults with selected clients to establish world-class service operations and is considered an expert in the area of process improvement, staffing models, and change management. Reach Dave at his office in Boulder, Colorado, at 303-494-4932 or [dave.brown@SupportCenterU.com](mailto:dave.brown@SupportCenterU.com).*