



6 ESSENTIAL SKILLS FOR TODAY'S PERFORMANCE ENGINEER

Softech Solutions Inc.

www.technosolution.tech

saney.alam@techsolutions.tech

6 Essential Skills for Today's Performance Engineer

Generating performance metrics is something that can be done relatively quickly and easily for many applications today. There are myriad commercial products that attack this problem from different perspectives. Regardless of how you generate test data and metrics, the same problem exists at test conclusion, "what now"?

Today's performance engineer is faced with a complex task when it comes to actually **improving** the scale or responsiveness of an application. Applications have grown much too complex and inter-dependent to simply throw more compute resources or bandwidth at the problem and expect good results.

Here are 6 essential skills being put to use by the best performance engineers in the business and how they can be applied:

1. UNDERSTANDING THE BUSINESS

It may sound cliché to state that engineers need to know the business but in this case it's true. Understanding the business impacts of the application being tested is important background to base recommendations. It's also very useful when vetting test scenarios and metrics being pushed to be tested by the applications team.

If an applications team says they have to have 10,000 simultaneous user registrations with 100ms response times for 5 hrs. straight, this may or may not be a valid business requirement. If this is extremely unlikely to ever occur then being able to set proper expectations with that team can limit unnecessary test cycles. More importantly, you will be in a position to potentially save the business money, eliminate pointless test cycles and save time trying to solve a problem not worth solving.

The converse is also equally valid. Many times a team new to web-scale applications may underestimate how quickly traffic and load can ramp up. An experienced performance engineer is in an excellent position to provide guidance and reshape the test scenarios to reflect real-world possibilities.

2. DOMAIN EXPERIENCE WITH WEB TECHNOLOGIES

Because we are talking about web performance, domain experience is the norm here. It may not be feasible to have developer level skills if that isn't your daily focus. Knowing the difference between a get and a post? A must. Same goes for how mobile apps are constructed and the latest trends in web development and delivery.

Knowing how to code beautifully efficient JavaScript might not be a No. 1 priority, but knowing what well-formed JavaScript code looks like certainly should be.

Qualified performance engineers typically come from web security, development or content management backgrounds. The web is the medium by which most applications (browser & mobile) are delivered, being up to speed here is key!

3. BROAD BASED IT KNOWLEDGE

It's really important to have at least a general understanding of the components when trying to solve problems with a complex system. With a web or mobile application this system is comprised of a number of moving parts and varies from company to company. To further add complexity, third party tools have proliferated dramatically in recent years.

The well-equipped performance engineer should be familiar with the internal components impacting performance alongside the external ones.

Internal technologies such as firewalls, load balancers and application delivery controllers, switching/routing, front end presentation servers, application servers, database servers and even mainframes all play a role. Digging even deeper in the actual code and configuration on many of these devices can adversely (or positively) affect performance and scale.

For example, on many anti-malware/next-gen firewall solutions there are a number of application level inspections. Many of these inspections are extremely processor intensive as they churn on user inputs to applications to determine if an attack is underway. A savvy performance engineer will be familiar with these systems and offer suggestions around feature tuning and other sources of potential improvements. No sense in having SQL database inspections on every session if there isn't a SQL database, right?

External components can become even more challenging to deal with as they are often out of the IT team's control. It is not uncommon to find and report problems with external ad networks, analytics or CDN solutions while performing an end to end performance test.

It's important for the performance engineer to recognize when there are issues outside of the IT team's control and make appropriate recommendations to mitigate them. This may come in the form of tactful communication with a third party or even changing application code to load objects in parallel, removing the need to wait on third party responses. This will have the effect of improving the end user perception of overall performance, which is very important.

4. NATURAL PROBLEM SOLVER

The fun part about this sort of work is uncovering and providing solutions to problems. Applications, requirements and environments will vary and so will the potential solutions. This keeps things interesting and fresh but isn't for someone looking for a boring 9-to-5 job!

If solving performance and scale issues were as simple as buying more bandwidth ([it's not as you'll see in this e-book by Load Impact's founder](#)) or adding more compute resources then we wouldn't be having this discussion.

A successful performance engineer will be a natural problem solver who can look at the big picture but also drill down into details of each component to uncover areas for improvement. As applications and infrastructure evolve so will the techniques used to improve performance. One thing that won't change, however, is the importance of problem solving skills.

5. PROJECT MANAGEMENT SKILLS

Improving performance is fast becoming an integral part of the DevOps process. No longer are a onetime piece of consulting work, this process is integrated into a continuous development cycle to ensure performance and scale thought about from application inception through the entire life cycle.

Being able to effectively project manage the testing phase of this is important for several reasons. Application development cycles can be very rapid and incorporating new processes such as load and performance testing cannot be a limiting step. Moving quickly and providing value will ensure performance engineering becomes a permanent part of this cycle.

Thorough testing and actionable recommendations is also critical. It won't do much good if the IT team can't take action and see measurable improvements during this process.

6. COMMUNICATION SKILLS

Arguably the most important skill in this list is effective communication. An engineer in this position has the responsibility to communicate with a number of stakeholders who all have their own priorities and areas of domain expertise.

As a relatively new field of work, it can be challenging to get others to accept the necessity to incorporate new testing procedures and recommendations. This makes tactful, effective communications even more critical to ensure stakeholders are on board with the process.

While many engineers are excellent at their area of expertise it is not as common to pair this with strong communication skills and tact. This is what sets highly successful technical consultants apart from others. It should be noted that regardless if you are an employee or outside consultant, proper communication strategies with the organization are important.

Performance testing is a field that will continue to grow rapidly and evolve — similar to the way our interactions with applications have evolved. Of course, these six traits aren't the only characteristics a performance tester should focus on, but they're definitely a good start!