

Performance testing

reducing risk



There are plenty of high profile stories demonstrating the importance of performance testing in reducing the likelihood of loss of reputation or revenue. In May 2012, during the Facebook IPO, the NASDAQ trading system folded under the volume of trading orders. This resulted in NASDAQ having to pay discomfited users \$63 million in compensation and fines of \$10 million to the Securities and Exchange Commission.

Other examples include the ticketing sites for the London 2012 Olympics, and more recently the Glasgow Commonwealth Games where both organisations failed to anticipate or cope with the level of demand with the Glasgow site being unavailable for over a week.

The cost of this kind of performance failure can be measured, not just in lost sales or financial penalties, but also in damage to reputation. As a result, almost all forward thinking organisations now realise the importance of performance testing and it is becoming a well-established part of a mature release process.

However, the adoption of performance testing does not necessarily mean that it is being done efficiently or effectively. Performance testing is a specialised process and not doing it correctly can result in costly and wasteful activity.

Nearly every round of performance testing requires performance requirements modelling and performance test design before the application is deployed onto the performance test environment.

This would also include the creation of the necessary test data, test script creation, test script modification, setup for test data collection (the amount of data that a well organised performance test will create should not be underestimated) and so on. Frequently, this work can require more time than the test execution itself. Therefore, when dealing with multiple releases of an application, careful decisions need to be made about when to test.

How to decide whether a release warrants performance testing

1. Is this the first time your users will use the application?

Regardless of how much confidence you have in the ability of the design and architecture of the application to sustain the load, there can be plenty of unforeseen and undetected traps that can trip you up. Additionally, the capacity for the users to use a new application in a completely unexpected way should not be underestimated. Internally, the feeling may be that the user load will be low, the application is over-architected and the infrastructure over-specified. But, as long as there are multiple users who will be using the application concurrently, there will always be a non-trivial risk that it could fail.

2. Are there likely to be spikes in the number of users or frequency of use?

If the user base is expected to increase or the usage spike significantly after a release, that is another prime reason for it to be performance tested. That spike in usage could be the trigger that exposes flaws in the system.

3. Does the release add functionality that involves extended processing or multiple calls between servers, or remote invocations?

Any user request that is passed along between multiple application layers, and involves back and forth or asynchronous communication between the layers, becomes a potential candidate for testing and verification. This is especially so if the change involves changing the size or the processing time significantly for any one of these calls. If a request makes four different calls to query the database before a response goes out to the caller, then an increase in one second for each of these queries could lead to a four second increase in the response time that the user experiences.

4. Have changes been made at any layer of the application that acts as a conduit for requests and responses?

Changes to configurations or settings on the primary firewall, switch, load balancer and so on are examples of conditions that meet this criteria. Changes at these layers might appear simple and straightforward to carry out, because in all likelihood they are being carried out on off-the-shelf products. However if they are configured incorrectly, they can slow the response behaviour of an application so much so that it becomes virtually unusable.

5. Have there been changes in the authentication mechanism?

Changes at the point where the user or session is authenticated are another key trigger for performance testing. Given that new security measures are recommended almost daily, it is quite possible that the authentication mechanism changes over time. Changes made at this point can have a significant impact on performance. For example, going from a non-secure HTTP connection to an SSL based secure connection could add as much as a 300% time overhead to a request-response round trip.

6. Have there been functional changes or improvements that involve complex processing?

Functional changes and improvements often involve multiple sequential calls to the database or complex query processing. Changes of this type have the tendency to not only slow functionality down for users but also, because of the effect they could have on database resources, impact the performance of other parts of the application.

Balancing the costs and benefits to achieve value from performance testing

Performance testing delivers important benefits:

- It reduces the risk of losing customers or revenue owing to badly performing software or applications
- It reduces the likelihood of costly reworks
- It ensures the best configuration of software, reducing the need to spend budget on more costly hardware
- It allows you to make comparisons between releases to understand performance trends and plan for future architecture, infrastructure and capacity
- It can help you meet your regulatory requirements



Performance testing is a highly specialised and potentially time-consuming activity and therefore needs to be targeted where it will provide best value.

At Capita, the due diligence that we carry out on each of our performance engagements ensures that testing is undertaken where it is necessary and valuable. We are able to deliver a range of performance options, from static performance reviews, through to load, performance and soak testing. This investigative and problem solving approach allows us to maximise your performance return on investment whilst reducing costs. You will get more from your project budgets and achieve more within your project schedules.

In our experience, engagement in the early stages of a project is also key to reducing overall costs. Early involvement allows us to influence the design and build, thus reducing the need to make expensive changes in the later stages of the project or after it has gone live.

Author

Aby Kurian - Performance Test Consultant

For further information about these services please contact:

Email: marketing.itps@capita.co.uk

Tel: +44 (0) 8456 077466

www.capita.co.uk/itprofessionalservices

