A Project Management Spotlight on Performance Testing [1], [2]

Bill Sundermann

During the software and operational challenges Southwest Airlines faced in late December, 2022, project management professionals likely asked themselves "Why wasn't the system tested to handle these use cases of heavy customer travel combined with the likelihood of bad weather impacting flying conditions?" Other airlines were able to recover quickly from the disruption caused by the frigid cold in Chicago, Denver, and other northern airports in the path of the storm.

The Dallas Morning News reported Southwest Airlines canceled more than 4,800 flights over two days as its operational meltdown continued into a second week with no indication of when it would end. Dallas-based Southwest canceled 2,509 flights nationwide on a Wednesday, about 62% of its schedule, according to Flightaware.com. The problems escalated, when the airline decided it needed to cut about two-thirds of its schedule over the coming days to try to reset the operation. Between December 22 and 28, the carrier canceled 13,353 flights, affecting as many as 2.3 million people. (1)

"The tools we use to recover from disruption serve us well 99% of the time, but clearly, we need to double down on our already existing plans to upgrade systems for these extreme circumstances so that we never again face what's happening right now," CEO Bob Jordan said in a video apology to customers during the disruption. "It almost became a running joke around the company that we aren't able to make certain changes because it would involve technology," said Lyn Montgomery, president of the union representing the 16,000 flight attendants at Southwest, TWU Local 556.

Southwest uses internally built and maintained systems called SkySolver and Crew Web Access for pilots and flight attendants. They can sign on to those systems to pick flights and then make changes when flights are canceled or delayed or when there is an illness. SkySolver and Crew Web Access are both available as mobile apps, but those systems often break down during even

[1] *Editor's note: Second Editions are previously published papers that have continued relevance in today's project management world, or which were originally published in conference proceedings or in a language other than English. Original publication acknowledged; authors retain copyright. This paper was originally presented at the 15th UT Dallas PM Symposium in May 2023. It is republished here with the permission of the author and conference organizers.*

[2] How to cite this paper: Sundermann, B. (2023). A Project Management Spotlight on Performance Testing; presented at the 15th University of Texas at Dallas Project Management Symposium in Richardson, TX, USA in May 2023; republished in the *PM World Journal*, Vol. XII, Issue XI, November.

mild weather events, and employees end up making phone calls to Southwest's crew scheduling help desk to find better routes. During periods of heavy operational trouble, the system gets bogged down with too much demand. In a Wall Street Journal article, Southwest said the disruption will reduce its pretax earnings by $725 million to $825 million. [(2)]

In another high-profile system failure in November 2022, the Ticketmaster sales platform was overwhelmed from high demand for Taylor Swift concert tickets. CNN Business reported that as sales for the singer's new Eras Tour began, heavy demand snarled the ticketing site, infuriating fans who couldn't buy tickets [(3)]. Customers complained about Ticketmaster not loading, saying the platform didn't allow them to access tickets, even if they had a pre-sale code for verified fans. On a Thursday, Ticketmaster announced that the sale to the general public, which was scheduled to begin the following Friday, had been cancelled due to "extraordinarily high demands on ticketing systems and insufficient remaining ticket inventory to meet that demand." The issues for Ticketmaster started earlier that week on Tuesday, when the site's sale kicked off for "verified fans" — a mechanism aimed at eliminating bots that gives presale codes to individuals. The "verified fan" platform was created in 2017 to help Ticketmaster handle situations of enormous demand (more than 3.5 million people pre-registered to be a Swift "verified fan"). That unprecedented demand, combined with a "staggering number of bot attacks as well as fans who didn't have invite codes" drove "unprecedented traffic" to its site, Ticketmaster said, and, essentially, broke it.

These are just two very widely reported instances, but there are predictably many more non-publicized instances every week of systems failing to perform as they were designed, promised, promoted, and contracted to do. The need for well-planned, thorough performance testing is increasingly entering into the risk assessments of development and upgrade projects.

Why Performance Testing is Important to Project Managers

Meeting performance requirements has a higher focus in the digital transformation age. Project Managers who are leading teams in developing software in-house or responsible for monitoring the efforts of external third-party vendors should be actively involved with planning and executing a successful launch or upgrade that includes performance testing.

In its Software Extension guide, the Project Management Institute (PMI) advises that project managers are tasked with providing leadership in roles and activities that involve product testing that could involve performance:

- Initiating, planning, and developing estimates initially and ongoing.
- Monitoring and controlling schedule milestones, budget expenditures, requirements stability, staff performance, resource utilization, and identified risk factors.

- Leading and directing by defining the project vision and maintaining it as requirements and other constraints change. Hands on, day-to-day leadership.
- Maintaining compliance with organizational policies and contractual requirements.
- Managing risk by identifying, analyzing, prioritizing, and responding to risk factors on an ongoing, continuous manner.
- Facilitating, coaching, monitoring, inspiring, and working with the software engineering knowledge workers to obtain desired results.
- Communicating with stakeholders to bridge the "technology gap" by using terms and concepts that are familiar to stakeholders.

The guide goes further to explain "Project managers are not expected to have the in-depth knowledge and skills of their team members, but they should understand the issues and concerns their team members deal with and should be familiar with the terminology used by the team members. Software project managers should also understand various approaches to managing software projects within the continuum of software project life cycles." (4). When followed, this guidance will help PMs become a more valuable part of any project team.

PMI leadership provides insight on skills aligned to performance testing. Pierre Le Manh, President and CEO of PMI, outlines skills project professionals probably need to cultivate in the third decade of the 21st century. Risk management is no longer just about weighing immediate project-related risks but assessing potential hazards in the broader enterprise. Data literacy is important. Data powers business these days. It underpins nearly all major decisions and is increasingly our most important asset. Project professionals must have at least a basic understanding of how to access and manipulate data. Even more important, they must be able to extract meaningful insights from data and communicate those insights confidently (5).

"Ignore the usual job titles: With the job market in such flux, it's important to focus on an organization's strategic needs rather than traditional job titles" says Michael DePrisco, COO of PMI (6). This means that an assigned project manager needs to be invested in the performance success and business value of a product.

In the Harvard Business Review Project Management Handbook, author Antonio Nieto-Rodriguez points out the increasing need for project managers to develop the qualities of project leaders who possess technical skills, product development and domain expertise, and strategy and business acumen. These competencies give credibility with the team and with stakeholders (including the testing team). They also enable communication in the language of the experts and the product teams; and provide a grasp of the project benefits and how and when they will be achieved (7).

Depending on the work environment, culture, and job expectations, Project Managers can be involved in reviewing and approving the testing strategy and test plans for the project. While the PM will not be directly conducting the tests, an understanding of the process, complexities, and required outcomes of performance testing is critical for managing a successful project.

With an ever-increasing reliance on technology to conduct business and to satisfy customers, projects will focus on ensuring that systems are at their highest level of performance and are scalable to meet growing demand. PMs must therefore advance skills to be transformational. Isaac Sacolick, author of Driving Digital and Digital Trailblazer, emphasizes that organizations need updated sets of practices to compete in a digital environment, deliver winning customer experiences, develop digitally driven products, and leverage data for strategic insights. For PMs, contributing to areas that haven't been worked on before are necessary for digital leadership [8] [9]. Knowledge of testing processes, and particularly performance testing, will enhance the contributions of the PM to the success of the project.

What is Performance Testing

Performance testing is distinct from functional tests in that it focuses on system attributes of responsiveness and stability under varying workload scenarios over a designated timeframe.

Typically, a performance test applies different increments of load and varying network conditions are introduced. The time taken for the system to respond is measured and evaluated. Speed, scalability, reliability, robustness, and stability of the application are all part of these load tests. These metrics can be defined as follows:

- Speed – Response time
- Scalability – Changes in response time as load (transaction volume, number of users) is increased.
- Reliability – Evaluates if all functions are consistently available under prolonged periods of high user traffic.
- Stability – Determines if the system remains available under heavy concurrent user load.

The high-level goals of performance testing are to determine if the system is production ready, capture the metrics of the tests mentioned above, identify any bottlenecks (software, hardware, network), and assess impacts. These goals are met by working through the incremental phases of performance testing, which should be included in the test plan:

1. Analyze, evaluate, and measure the existing system.
3. Develop the required test assets.
4. Create and automate reusable test scripts.

5. Establish and configure the load testing environment.
6. Execute benchmark tests, analyze results, and document plans for improving performance.
7. Publish reports for tests run and a final closure report.
8. Obtain management signoff of the performance testing phase.

Most performance testing is done to measure simulated web traffic or simulated data processing capabilities.

Tests that simulate web traffic – In a performance testing environment, these tests simulate virtual users to check how well the system performs under load. These include stress, load, endurance and spike tests.

Tests that simulate data processing capabilities – This performance testing measure aims to test how the system handles the constant influx of new data batches. These include soak and scalability tests.

Common types of software performance testing fall into these categories as described by Testing Experts [(10)]:

- Load testing: Load testing is performed to validate the system (application under test) performance under normal (usually around 70% of peak user load) and peak user load. This type of test helps us to tune the system and finalize the baseline. Companies should adopt load testing as a part of their software development life cycle (SDLC).

- Stress testing: This type of testing is done to identify the breaking point of the system when its subjected to a user load beyond the expected peak. The breaking point can be defined as a load at which either the response time degrades beyond 10% or transactions start falling by more than 5%.

- Spike testing: This is a subset of stress testing, and it analyzes the behavior of the system by suddenly varying the number of users and it also checks if the system is able to support such situations efficiently.

- Endurance testing: It is a non-functional type of software performance testing and typically checks the behavior of the system when it is under significant load (around 70% of peak load) for a longer period of time (usually 8 hours to 72 hours). Specifically, the system is checked for resources leakage like memory leaks, system failure, or any other random behavior that might be shown by the system.

- Soak tests - Soak tests gradually increase the number of virtual users and check how well the system handles loads over a more extended period. The main objective of these tests is to check if sustained high user activity over longer time periods negatively influences performance levels.

- Scalability (capacity) tests - Designed to test the ability of the system to process more data the more hardware resources are dedicated to it. A key thing to know for CTOs is whether their system is capable of scaling out horizontally.

Performance testing may mean different things to different companies, depending on the focus of their applications. Regardless, two key metrics are basis for any performance measurement – latency (the delay between a user takes an action and response) and throughput (the number of such actions executed or results produced per unit of time).

Successful project management is dependent on following proven processes that adhere to advanced planning and monitored execution. Even agile methodologies, focusing on interactions and response to change over processes and plans, have at their core the need to plan a course of action in advance. This extends to the area of performance testing where definition of non-functional requirements, infrastructure buildout, and adoption of adequate tools in the early stages are keys to success.

A Short Case Study on Performance Testing

In the 2nd quarter of 2019, FIS was nearing the full marketing push of its Modern Banking Platform, a new product offering in the financial services industry. The platform represented the culmination of planning and constructing a next generation, core-banking system built from the ground up using service-oriented, component-based, API first architecture that could be deployed to the cloud using container managed technologies.

As the Project Manager for the Enterprise Architecture Team, I was tasked with planning and organizing the performance testing of the platform. Dell Technologies had recently been contracted to upgrade FIS Data Centers. As a benefit of this relationship, Dell offered to provide the Dell Lab in Round Rock, TX as the environment for the FIS tests. After FIS derived some high-level estimates of account volume, transaction mix, and performance objectives, the engineers at Dell prepared recommendations for Database Servers, Applications Servers, and CPUs for Application and Database Tiers in the testing environment to fit overall processing requirements.

Performance testing had been informally conducted by the DBAs on the development team, but there had been no structured, goal-focused approach and execution was outside of the regular functional test cycles. The first priority was to develop a test plan for the Dell Labs including the test objectives, the infrastructure (including hardware diagram), and technology stack. The test plan also included specific information on the metrics to be captured, how the results would be documented, the communication channels, how the results would be archived for comparative analysis by cycle, and establishment of a change log. The Performance Server Configurations were organized by tabs to include server details, container details, the transaction mix, and calculations used in metrics such as response time, effective TPS (Transactions Per Second) per User, and Client Threads.

With the Dell Lab being outside of the FIS Data Center control, technical resources had to be enlisted to deploy the containerized applications, configure the servers, establish load balancers, and help support the initial cycles. As the performance tests were underway, several action items were identified including providing the location for all daily documentation, providing other team members ability to run tests independently and have access to logs, monitoring the AWS reports in real time to understand issues, and arrange for licenses for a database optimization tool (DBOptimzer). Other issues encountered were environment eviction after 2 hours and buffer cache buildup, enabling the Oracle huge space parameter and memory targets, and garbage collection adjustments to address disk pressure issues.

After two months of testing, the team met to brainstorm on potential causes of bottlenecks encountered. It was concluded that bottlenecks were not on the database, but from an application perspective. There was not an issue with pushing the load, but a matter of the load getting into the system. One adjustment was in the approach of starting testing at an upstream point to force errors on load generators (to know threshold) and then work downstream to identify bottlenecks down the stack. Another expanded strategy was to measure saturation points at every stage of the flow from load injector → load balancer → service → data source pool → database (connections) → storage.

Primary lessons learned were to get leveraged services committed early, use an approach that tests incremental flows, and obtain any need licenses sooner in the calendar.

Performance Testing Best Practices

Project Managers with responsibility of delivering projects on schedule must be knowledgeable in the testing phases to plan milestones, manage risks, and communicate progress. Increasingly, performance testing is a critical part of testing activity.

Netreo, an award-winning provider of IT infrastructure monitoring and observability, provides these performance testing best practices (as described in product offerings for Stackify application performance monitoring software) [11]:

- Test as early as possible in development. Do not wait and rush performance testing as the project winds down.
- Performance testing isn't just for completed projects. There is value in testing individual units or modules.
- Conduct multiple performance tests to ensure consistent findings and determine metrics averages.
- Applications often involve multiple systems such as databases, servers, and services. Test the individual units separately as well as together.
- Involve developers, IT and testers in creating a performance testing environment.
- Remember real people will be using the software that is undergoing performance testing. Determine how the results will affect users not just test environment servers.
- Go beyond performance test parameters. Develop a model by planning a test environment that takes into account as much user activity as possible.
- Baseline measurements provide a starting point for determining success or failure.
- Performance tests are best conducted in test environments that are as close to the production systems as possible.
- Isolate the performance test environment from the environment used for quality assurance testing.
- No performance testing tool will do everything needed. And limited resources may restrict choice even further. Research performance testing tools for the right fit.
- Keep the test environment as consistent as possible.
- Calculating averages will deliver actionable metrics. There is value in tracking outliers also. Those extreme measurements could reveal possible failures.
- Consider the audience when preparing reports that share performance testing findings. Also, include any system and software changes in reports.

Finally, express gratitude to your team and others assisting your project. A recent experiment found people spent 15% more time helping someone who thanked them for their prior help versus someone who did not. [12]

Perhaps the most important best practice is to test early, also known as shift left. A shift left approach brings development and testing together early with the intention of finding and preventing defects early in the software delivery timeline. To support this, Tricentis, a global leader in enterprise continuous testing published a whitepaper stating one of the primary challenges they encounter with their clients is the difficulty in keeping up with the pace and scale of development teams so that feedback regarding performance can be received early and more frequently [13]. With the fast pace of software development-test-release cycles, it is imperative that teams anticipate and solve performance issues early in the SDLC. Test automation and a continuous testing methodology is now used by many teams in a DevOps approach ensuring development and operations collaborate over the entire cycle. The aim is to accelerate software delivery while balancing cost, quality and risk. With this testing technique, teams don't need to wait for the software to be built before testing starts. They can run tests much earlier in the cycle to discover defects sooner, when they are easier to fix. Tricentis provides guidance on maximizing the value of test automation:

- Define action to be taken - Automated testing is only beneficial when it produces actionable steps from reliable testing
- Prioritize which performance tests are in the critical path and need continuous feedback. The method for prioritizing can be based on the USE Model – utilization, saturation, errors.
- Automation test planning requires a clear statement of non-functional requirements and criteria, clearly communicated goals and outcomes (e.g. SLAs), and providing for a test environment that is consistently available.

Performance Testing Tools

Project Managers should ensure test plans include the most current tools for testing software performance. Gartner classifies these tools as Application Performance Monitoring and Observability [14]. Gartner defines the application performance monitoring (APM) and observability market as software that enables the observation and analysis of application health, performance and user experience. The targeted roles are those the Project Manager is closely working with: IT operations, site reliability engineers, cloud and platform ops, application developers, and product owners. These solutions may be offered for self-hosted deployments; as vendor-managed, hosted environments; or via software as a service (SaaS).

The capabilities of APM and observability tools include:

- The observation of an application's complete transactional behavior.
- Automated discovery and mapping of an application and its infrastructure components (including cloud services).

- Monitoring of applications running on mobile (native and browser) and desktop browsers.
- Identification and analysis of application performance problems and their impact on business outcomes.
- Native integration capabilities with automation and service management tools, as well as native integration with public cloud providers — e.g., Amazon Web Services (AWS) Cloudwatch, Azure Monitoring and Google Cloud Operations.
- Analysis of business key performance indicators (KPIs) and user journeys — for example, login to check-out.
- The ability to perform interactive interrogation of multiple telemetry types (such as traces, metrics and logs) to detect “unknown unknowns” — that is, the ability to identify underlying issues to unexpected events and gaps in telemetry coverage.
- Application security functionality delivered via a common agent or framework for APM.

Optional functionality may include:

- Endpoint monitoring to understand the user experience and its impact on business outcomes.
- Support for virtual desktop infrastructure (VDI) monitoring.
- Performance testing and integration with load testing tools.

The capabilities of these tools highlight the importance of their adoption in the planning phase of the project. As part of Test Plan approval, the Project Manager should observe that the testing teams are equipped to evaluate these areas of non-functional performance.

Steve Tack, SVP of Product Management and Dynatrace notes “As the market has moved to modern hybrid and multicloud environments, the data and complexity generated by these clouds have grown exponentially” (15). This growing complexity is the key reason PM’s need to be involved at the appropriate level to ensure success of the overall project. Tack goes on to say “Our focus on delivering precise answers and intelligent automation from data has enabled our customers to do their clouds right, minimizing cloud complexity, accelerating their adoption of cloud-native technologies, and speeding digital transformation.” The Dynatrace release on Businesswire stated “observability from the Dynatrace platform extends beyond metrics, logs, and traces to user experience data, runtime security data, and data from the latest open-source standards, such as OpenTelemetry. Davis, the AIOps engine at the platform’s core, processes this data in real time, monitors the full stack for system degradation, performance anomalies, and security vulnerabilities, and delivers precise answers prioritized by business impact. In addition, the platform automates error-prone, manual tasks, including continuous discovery, proactive anomaly detection, and optimization across the software lifecycle, to dramatically reduce routine manual tasks and enable DevSecOps teams to ensure flawless and secure digital interactions."

The Project Manager who continues to develop skills in data literacy, as PMI President and CEO Pierre LeManh said in his blog, will be in the best position to maximize their value. Becoming knowledgeable with the capabilities and applications of performance testing tools is a great asset to the project team.

AI in Performance Testing

Most businesses know that in order to adapt to rapid transformation in the digital world, they must develop products and services based on advancing technology and undertake projects that can be delivered quickly. To test these increasingly complex deliverables, Artificial Intelligence (AI) can add significant value to the process – especially in the area of performance testing.

Project Managers should begin any planning with recommendations that the testing team add AI into test planning for daily tasks such test design and scripting. Performance test modelling can benefit from AI pattern recognition that can anticipate future load problems. AI analysis of system complexity can identify bottlenecks in multiple tiers of the technology stack in early stages of development.

With advancements in Artificial Intelligence (AI) and Machine Learning (ML) powered software, during the early stages of application design, Project Managers should receive answers to these questions posed to performance engineers: Does the expected level of performance once the application is in production align with our requirements from the customer? Have we identified the potential bottlenecks? What is the expected effort for the engineers to tune application parameters during testing cycles?

AI is the intelligent part of the Performance Testing process that adds maturity to scripting, execution, and monitoring. It acts as brain in the process. AI is also helpful in filtering and reporting the key metrics of poor performance, giving the project team time-saving information on where to focus efforts.

By implementing AI into the test plan objectives, specific challenges can be mitigated or eliminated by:

- Having clearly defined non-functional requirements.

- Developing workload modeling that expedites testing.
- Simulating the user journey into test scripts involving complex technology.
- Deriving meaningful and useful test data sets.
- Identifying bottlenecks
- Producing dependency maps to assist with root cause analysis.
- Utilizing rule-based code analyzers to predict coding issues, thereby reducing time required to resolve problems.

In November 2022, OpenAI released ChatGPT (Generative Pre-trained Transformer) which has sparked a high level of interest from Project Managers seeking to capture the promised benefits of assisted project plan development, automation of task definition, scripting of test cases, solving for multiple dependencies, and organizing data for reports. While still in its early stages but already growing to 100 million active users, the chatbot is another AI tool that will become more pervasive in the project management toolbox. With this increased use, PMs will need to be adept at formulating prompts to effectively utilize the AI language processing capabilities.

Conclusion

Performance testing has been raised to the forefront of project management in the age of digital transformation and rapid technology deployment. Project managers should be taking a leadership role in ensuring the project requirements reflect any performance goals of the customer and should integrate these specifications into the planning process, the scope statement, and the project objectives. Finally, completion of performance testing that satisfies customer requirements should be part of any project exit criteria.

References

(1) Ferguson, Lana; Dallas Morning News kyle.arnold@dallasnews.com

(2) The Wall Street Journal, *Southwest Airlines to Revamp Crew Scheduling System*
WSJ Jan. 12, 2023 7:25 pm ET

(3) Frank Pallotta, CNN Business, Updated 8:53 PM EST, Fri November 18, 2022

(4) PMI Software Extension to the PMBOK Guide Fifth Edition

(5) Pierre Le Manh, President and CEO of PMI, The Top 10 Project Management Skills You Need in Your Toolbelt. The Official PMI Blog, October 5, 2022

(6) Michael DePrisco, COO of PMI, Survival Tips for a New World of Work, The Official PMI Blog, June 23, 2022

(7) Nieto Rodriguez, Antonio. *Harvard Business Review, Project Management Handbook, p.205.* 2021 Harvard Business School Publishing Corporation.

(8) Sacolick, Isaac. *Driving Digital*, 2017, American Management Association

(9) Sacolick, Isaac. *Digital Transformer*, 2022, John Wiley and Sons

(10) Performance Testing Categories testing experts: https://www.testingexperts.com/blog/performance-testing-guide/

(11) Stackify https://stackify.com/ultimate-guide-performance-testing-and-software-testing/

Additional Resources on Performance Testing

- Performance Testing Guidance for Web Applications (Microsoft)
- "Perfect Software and Other Illusions About Testing" (Book by Jerry Weinberg)
- Top Load Testing Tools: 50 Useful Tools for Load Testing Websites, Apps, and More (Stackify)
- The Tester (Special Interest Group In Software Testing (SIGIST) newsletter)
- Software Performance Testing (TechTarget)

(12) Algoe, Sara. "Why It's Important to Show Gratitude and Work – and What's the Best Way to Do It"; Wall Street Journal; 4/17/23.
(13) Tricentis Whitepaper: A Practical Guide to Continuous Performance Testing; 2021
https://www.tricentis.com/resources/continuous-testing-framework

(14) Gartner Magic Quadrant for Application Performance Monitoring and Observability
Published 7 June 2022 - ID G00750730
https://www.gartner.com/doc/reprints?id=1-2A8Q59D0&ct=220608&st=sb

(15) BusinessWire June 13, 2022 12:42 PM Eastern Daylight Time
https://www.businesswire.com/news/home/20220613005720/en/Dynatrace-Named-a-Leader-in-Gartner%C2%AE-Magic-Quadrant%E2%84%A2-for-APM-and-Observability

About the Author

Bill Sundermann

Texas, USA

Bill Sundermann is a certified PMP with 27 years of experience in the financial services software industry. As a senior project manager, he is passionate about training and sharing knowledge in the areas of agile coaching, team building, and effective messaging. He can be contacted at bsundermann@verizon.net