



## THE 20 BEST 2018 PERFORMANCE TESTER INSIGHTS

**2nd edition**   **20** speakers from **10** countries   **24** hours live

**35** cup of coffee   **800** registrants   **950** stickers

**This past June, Neotys brought together 20 experts from around the world to share their experience and knowledge regarding techniques, challenges, and best practices associated with load testing for the modern enterprise.**

**The gathering was delivered online as a virtual event.**

**The second annual Performance Advisory Council (PAC) event, proved to be transformational. Attendees were treated to new and innovative practices on the art and science of performance testing. Each presentation provided a wealth of information with insights and techniques aimed at performance testing improvement.**

**The following is a synopsis of several presentations made during the event. We're happy to share this information in the spirit of creating a more well informed, efficient community of test practitioners. Enabling user story sharing and learning on an international basis is critical to us.**



## Stijn Schepers

### Performance Testing is Not an Average Game!

[Stijn Schepers](#) has over 18 years of international experience as a performance test specialist/manager using both Agile (Scrum, SAFe) and Waterfall models. He has worked in Europe, New Zealand, and Australia as a performance engineer testing and optimizing complex enterprise solutions like Oracle: Financials, Forms, Web Center, CRM Applications, JD Edwards, J2EE solutions, Vignette, SIEBEL, SAP, Cúram, IBM Websphere, and Mule. Schepers is a visionary leader with in-depth technical knowledge in concept and practice driving performance testing in the modern enterprise.

Stijn feels that performance testing is about looking at “all the dots (raw data)” and that it’s not about seeking averages.” To him, raw data is what you want to measure – every single request that every single user is making because that gives you the insights you need to understand system behavior. He points out the variety of insights can be determined from the data gathered and that when testers focus on average behavior, they will miss important problems in the system. A broader analysis is needed.

In his presentation, Stijn applied his thinking in a variety of demonstrative examples using data exported from NeoLoad for visual analysis in Tableau. As the takeaway, he wanted his audience to make sure that any tool being used to conduct tests, allows the output of the raw test results data – without this, test practitioners will be significantly limited.

For the full session recording, click [here](#). You can also find [Stijn Schepers' blog](#) on the topic [here](#).



## Stephen Townshend

### The Forgotten Art of Log Analysis

[Stephen Townshend](#) is a software performance specialist based in Auckland, New Zealand. According to him, performance is more than just testing; it's about business risk. His primary job responsibilities are to identify, prioritize, and manage performance risk uniquely based on the particular customer's unique situation.

Throughout his hands-on presentation, Stephen talked about a "divide and conquer" approach. He demonstrated how to parse and interpret different types of system logs, paying close attention to web server access, platform, and database table log examination to understand and isolate performance bottlenecks.

Stephen narrowed in on log analysis to ensure system stability. He described ways of using log analysis to get useful results with both modern and traditional tools diving into various types of information that can be gleaned from the logs.

- Request
- Event
- Throughput
- Timing
- Error

Stephen emphasized the importance of gathering intelligence from team members about the purpose and meaning of standard logs as well as those which are unique to the enterprise. According to him, "You've got to talk to people to understand what the logs are telling you."

The presentation also included a review of the four operational guidelines for analyzing log data:

- Sample Size: "Making sure you have enough data to make informed decisions."
- Percentiles: "Not just looking at the percentile in isolation. You need to also look at the raw data."
- Raw Data: "Scalars are not very good at showing density (occurrence of values). This is when averages, percentiles, and sum or values come in handy."
- Multiple Angles: "Stop looking at data from a single lens. Don't just take an average and think this is "my answer"."

Stephen's presentation transcends the theoretical. He successfully articulates how to best compose a log parser in Python. Finally, he brings his content full-circle through demonstration – how to visualize performance test data in [Tableau](#).

For the full session recording, click [here](#). You can also find [Stephen Townshend's](#) blog on the topic [here](#).



## Srivalli Aparna

### Databases Done Right

[Srivalli Aparna](#) is a Senior Test Consultant for The Testing Consultancy, New Zealand. Over the last ten years, she has worked for some UK and New Zealand based clients helping them to improve their software performance. She recalls that one of the best learning experiences thus far as having been a part of the product engineering group at Oracle learning about database architecture.

The central point of Srivalli's PAC presentation – the importance of production system emulation testing as much as possible. For example, executing a performance test against a database with only a hundred records will produce different results compared to a production database of thousands.

Additionally, the data in the test database, as well as the DB structure, need to emulate the production system closely. Making sure that sharding, for instance, is similar and that the physical distance between the client test system and the test database closely matches the physical gap between calling clients and the database in production.

For the full session recording, click [here](#). You can also find [Srivalli Aparna's](#) blog on the topic [here](#).



## Thomas Rotté

### How AI based Business Intelligence Helps Test Automation

[Thomas Rotté](#) is CTO at Probit. His mission is to help application managers, and product owners of intensively used web applications reduce costs and increase quality by creating, maintaining, and executing automatically generated scenarios based on production behavior.

Thomas uses his experience to evangelize solutions while working on a user-analytics system combining Dynatrace data with AI to gain more insights and deliver business value to improve the way test automation is run.

Thomas doesn't discount the difficulty of testing and is quick to remind how expensive it can be. Hence, the practice of using AI as a sensible approach to time and cost reduction. In his session, he demonstrated how to use AI to segment user activities into different behavioral clusters. Also, displaying how to use a conventional, third-party analytic and monitoring tool such as Dynatrace to extract the data that gets absorbed by the AI algorithms. He concluded with a review of Probit, the product his team has developed, as means by which he gathers intelligence about user behavioral patterns.

For the full session recording, click [here](#).



## Ramya Ramalinga Moorthy

### Continuous (Early) vs. System Level Performance Tests

[Ramya Ramalinga Moorthy](#) is a passionate performance analyst with 15+ years of performance testing, modeling, application capacity planning experience. She is a computer science engineering graduate with a Masters in Software Systems (MS) from BITS PILANI, India. She runs an online academy (<https://elearning.qaelitesouls.com>) for performance engineers. She is also a prestigious award winner of the 2017 Computer Measurement Group's Bill Mullen Award for work on anomaly detection.

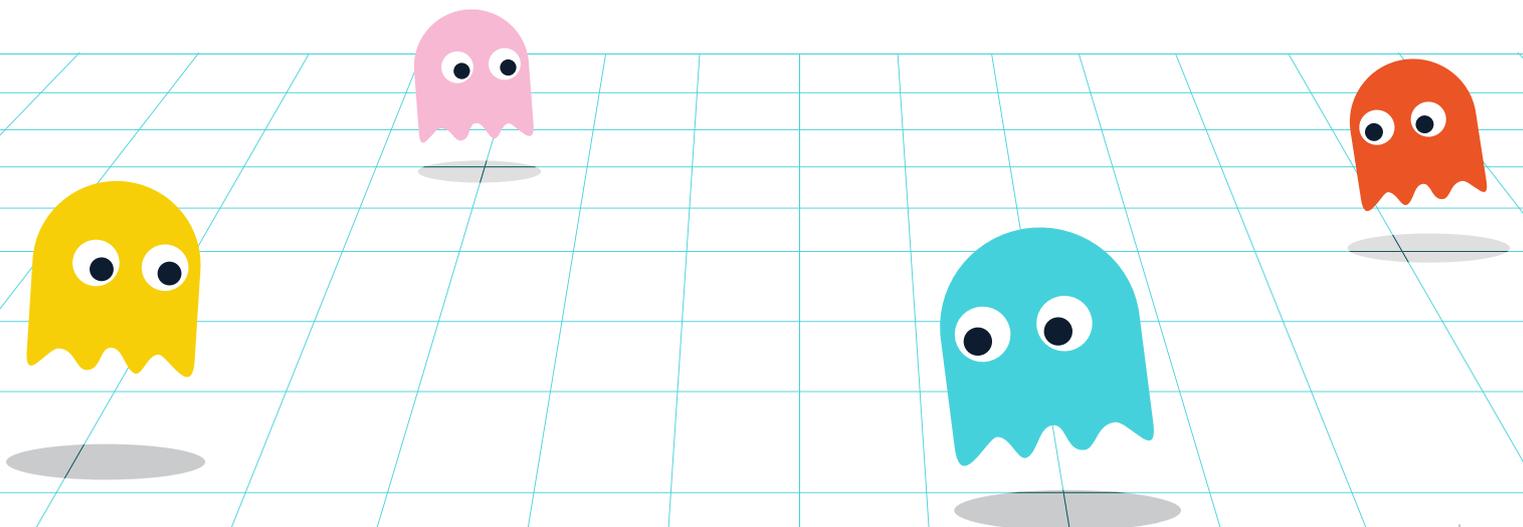
Ramya's PAC presentation highlighted the best approach for performance testing strategy. In doing so, she provided a list of do's and don'ts when testing within a CI/CD process as opposed to system level testing.

She confirms that a good performance testing strategy includes following best practices and techniques relative to Non-Functional Requirements (NFR) identification and verification. In turn, reinforcing the importance of test types, objectives, use case identification, infrastructure requirements, and KPI analysis.

Ramya argues that performance testing focus is a must have for DevOps cultural change success. To her, there are seven points to consider.

1. Maintain good testing objectives that are defined based on well-known Service Level Agreements (SLAs).
2. Include a performance analyst as early as possible within the scrum team.
3. Define when/what to test from the getgo – e.g., CI performance/load testing and system level performance testing.
4. Document automated test results (pass/fail criteria: CPU, response times).
5. Ensure that the performance test strategy describes what to test according to the environment.
6. Allocate enough time during test planning to select the right performance testing tool.
7. Provide sufficient time during the planning process to choose the proper Application Performance Monitoring (APM) tool.

For the full session recording, click [here](#). You can also find [Ramya Ramalinga Moorthy's](#) blog on the topic [here](#).





## Amir Rozenberg

### Shifting Left Performance Testing

During this PAC, Amir Rozenberg was the Director of Product Management at Perfecto. He has extensive experience in the digital industry with expertise in areas including application development, testing, delivery, and monitoring. Rozenberg has led successful engagements with Microsoft, Neotys, HP, and others to optimize value delivery to customers. He also founded Adva Mobile, a direct-to-fan mobile marketing startup.

Rozenberg asserts that there is a fallacy in the conventional thinking about non-functional testing, such as performance, accessibility, and security testing. The traditional thinking is that it's impossible to conduct non-functional testing during an Agile sprint. This is not the case. The reality is that parts of performance testing can be shifted left into the sprint, automated in such a way that responsiveness defects are exploited within minutes of code commit, independent of the platform.

In his PAC presentation, he defined digital transformation as “an expansion of digital interfaces that touch our end-users whether it's a B-to-B or B-to-C, on an enhanced, expanded number of screens/types (e.g., mobile, phones, tablet, things in between like IoT device and motor vehicle screens).”

Amir's review pointed out the expansion of the digital interfaces and experiences and how they put increased pressure on companies to bring new features to market quickly and to differentiate new interfaces and experiences in the marketplace. Also, now that clients are becoming more powerful, performance testing must put as much focus on client-side testing the same way it is on the server side. A result, companies are turning to Agile to increase productivity and decrease time to market.

To him, adopting Agile means conducting the testing activity, including performance testing earlier in the development cycle – the very intention of the Shift Left movement.

Amir's presentation included answers to questions regarding the viability of implementing non-functional testing early in the development cycle. According to him, some things can't be moved into the sprint. For example, manually intensive performance tests requiring fully functional/ integrated systems. However, responsiveness, accessibility, and security tests can be shifted left into the sprint. He highlighted the importance of implementing proper logging throughout the process and adequate low-level monitoring of machine and network activity.

Persona definition is also important in that responsiveness testing needs to happen on clients as they are in the “real world.” A well-defined persona definition will describe a client device environment in which applications other than the one under test runs in the background (having Pandora concurrently with the application under test). As Amir pointed out, all tests need to run in an environment that emulates the true state of the production circumstance.

For the full session recording, click [here](#). You can also find Amir Rozenberg's [blog](#) on the topic here.



## Thomas Steinmaurer

### Continuous Cluster Performance Validation @ Dynatrace

[Thomas Steinmaurer](#) brings his 15+ years of relational database technology experience to his current role. Having been involved from day one with the building of the new SaaS and On-Premise next-generation APM solution (primarily responsible for the Cassandra backend), he now turns his focus on load testing, performance, and scalability aspects of the Dynatrace Cluster on a daily basis.

In his PAC presentation, Steinmaurer pointed out that in the past, Dynatrace would deliver two releases per year. Today, he confirmed, Dynatrace has been known to produce 26 releases to production throughout a two-week sprint! He summarized that two factors allowed Dynatrace to increase the velocity of release.

1. Agile adoption
2. Testing philosophy adjustment

From here, Thomas dove into great detail about Dynatrace's architecture, describing how the company approaches, Continuous Delivery and Feedback (CDF). Dynatrace takes a three-phased approach to release.

1. Development
2. Acceptance
3. Production
  - a. Notes:
    - i. Typically, the development runs for two weeks as does acceptance.
    - ii. Both development and acceptance implement rigorous performance/load testing.
      1. Development testing activity is conducted daily.
      2. Testing in acceptance occurs every two days. Comprehensive monitoring is built into both phases.
        - a. The result is bi-weekly releases to production.

Dynatrace uses emulation extensively in testing, particularly about emulating tenant (customer) load. The emulations range from smaller tenants to substantial tenants. Keeping emulations close to the real world provides the reliability that Dynatrace needs to have confidence in its testing practices. However, when it comes to regression testing, Dynatrace uses a deterministic, unchanged load.

Dynatrace does not use a standard load testing solution such as JMeter. Instead, the company built their own solution to simulate precisely the load aligned with their agents and user interactions.

Dynatrace manually triggers the updates of the large-scale clusters to allow humans to monitor the condition of downstream activity for operational anomalies before pushing releases up. Still, the company does experience issues in Production. The lesson learned – that while doing proper emulation and comprehensive testing downstream before releasing to production is critical; problems will occur in the Production environment. Above all else, he recommended, that companies need to remain vigilant.

For the full session recording, click [here](#).



## Bruno Da Silva

### Continuous Performance Innovative Approach

[Bruno Da Silva](#) joined Leanovia after completing his studies at the UTC in Compiègne, France. He is passionate about IT performance testing, which is something that his current role allows him to follow every day. Bruno is certified in both NeoLoad and Dynatrace technologies.

Bruno's PAC presentation pointed out the three fundamental drivers of performance: Application, Load, and Resources. Application is the logic that drives behavior. Load, the users – both automated and human – who interact with the system. Resources are the hardware and network that support the system.

According to him, the goal of continuous performance testing is to control the robustness of the code. He reminded all of some of the many benefits to continuous testing and how they add up to application performance improvement.

- Cost reduction
- Risk mitigation
- Code improvement
- Automation
- Monitoring

Bruno also noted some prerequisites for continuous performance testing:

- Definition of performance requirements earlier
- Ensuring the testability of the software
- Having a production-like pipeline
- Creation of a sound plan

He listed the way that tools can be used to alleviate unnecessary work labor during continuous performance with a review of the tools that can be applied:

- CI/CD industrialization with the Jenkins pipeline (Blue Ocean, Git, SonarQube, Nexus)
- Containerized architecture with Openshift
- Monitoring from Dynatrace
- Performance testing with NeoLoad

Bruno's main leave-behind was that you shouldn't be afraid to take advantage of every tool's capabilities (plugins, APIs, and Artificial Intelligence) to speed up your performance validation process.

For the full session recording, click [here](#).



## Bruno Audoux

### Autonomous Scripts

[Bruno Audoux](#) has been writing code since the age of ten. He followed his computer technology passion by working in fluid data and computer architecture at Control Data, France. He started his company, Bench, in 2002 to create products for the French tax web services. In 2010, he created the company Aerow Performance (now WAPSI), where he is now a technical manager.

The focus of Bruno's presentation was to share scripting "tricks." His primary message was that modern scripts might need a shelf life that outlives a single sprint. His recommendation – use small scripts that can be aggregated to meet the need at hand. Much easier to manage than large scripts, a small one should be limited to a single transaction – e.g., a login process.

He continued with a demonstration of some aspects of scripting in the modern enterprise, covering more tricks to facilitate data exchanges between scripts via CSV file, a database, and a shared queue.

For the full session recording, click [here](#).





## Helen Bally

### Sub-second Response Time in Warehouse Management

[Helen Bally](#) started her IT career as the first SAP trainee for Price Waterhouse in London. SAP implementations have taken her to Ireland, Houston, and Brussels where she led global SAP technology projects in the Oil and Gas industry. Arriving in Switzerland in 2003, Helen joined Itecor and engaged with the world of testing. Today, she leads the Itecor Enterprise Testing practice and has helped numerous Swiss and International companies with test strategy, test automation, and performance testing. She has a love for complex technical projects where she can drive results by bringing together technology, people, and processes.

The focus of Helen's presentation was about how a significant manufacturing and distribution company achieved their goal of sub-second response time for warehouse operators. Her talk showed how the company implemented an SAP Enterprise Warehouse Management System (SAP EWM) with the hope of ensuring that all warehouse operator functions would achieve the sub-second response time goal.

While most performance testing is concerned with testing concurrent users, Helen's case was different. Her focus was on a single user with micro-analysis on every user click during real-world conditions.

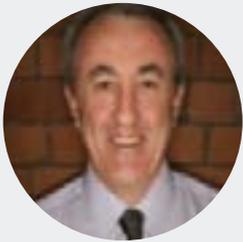
Her PAC presentation included a review of the unique challenges associated with the warehouse management scenario. For example, the state of the warehouse (e.g., how full it was) and its impact on performance.

Helen's team took an innovative approach to testing. They isolated the test environment and implemented the use of virtualized services using SAP RFC ([remote function call](#)) technology in automated test scripts (remote function calls). The test script took traces to analyze performance at the code level. Using their repeatable test framework, the team was able to prove that performance was improved.

The team learned some valuable lessons from experience:

- The setup/isolation of the service under test counts
- Testing success is dependent upon having great specs
- The test environment needs to be completely independent of ongoing warehouse activity so as not to affect day-to-day business
- You should make sure that the items you're stocking in the warehouse are representative of the real-world inventory
- Tracing, even within an ERP, is critical
- The scope of testing needs to monitor the performance to detect regression as well as compare it with a baseline
- Performance testing tasks need to replicate the real-world activity precisely one and one

For the full session recording, click [here](#).



## Ian Molyneaux

### Maintaining Application Performance as part of Digital Transformation

[Ian Molyneaux](#) is a Senior Consultant at Infuse. His landing in IT was purely by chance after applying for what he described as an “interesting” job as Junior Computer Operator in the mid-70s. 36 years later, Ian is still in IT, drawing from all of his experiences with a particular interest in application performance.

Ian is a published author on performance testing and related disciplines (The Art of Application Performance Testing, O’Reilly 2009 and 2014). He currently focuses on offering advice and support to clients on topics such as DevOps implementation, performance testing and troubleshooting, and Application Performance Management (APM). A regular industry speaker, you can also see Molyneaux working in an advisory capacity with many IT tooling vendors.

His PAC presentation portrayed the digital transformation as a three-part evolution.

1. A trend among enterprises to move to the cloud to realize cost and performance benefits.
2. A solution re-architecture to adjust to a new, cloud-based computing environment.
3. A user experience focus with supportive technologies that enable migration to cloud-based, distributed environments.
  - a. Notes:
    - i. Mobile devices that rely on cloud access to operate bring a whole new dimension to UX.

Today, because of the Internet of Things (IoT), mobile devices are moving beyond the cell phone and digital tablet. New types of interaction have become part of the user experience, for example working with a voice-based device such as Alexa. The dynamics of UX are intrinsically different; thus the approach to performance testing must be changed. According to Molyneaux, “the user experience is a huge part of this digital transformation process.”

Molyneaux also pointed out that each part in this evolution comes with its own set of performance risks. Therefore, when considering how to approach performance testing regarding a digital transformation, it’s best to focus testing activities on a single application in the digital transformation. “Don’t go big bang.” Confine the risk scope. Also, set a baseline of data for pre/post-comparison application performance in the cloud migration. Lastly, when making a digital transformation to the cloud, many companies turn over operation and maintenance of an application to a third-party. To ensure safety, they need to have a disaster recovery plan in place should things not go as planned.

When it comes re-architecting, Ian observed that cloud technologies are subject to a high degree of choice and change. “There are some approaches a company can take to designing a Service Oriented Architecture (SOA), and there are some products that can be applied to a given design.” For example, for container orchestration, there’s Kubernetes, Mesos, or Docker Swarm. Also, the backing data storage technologies can vary from traditional databases to those based on NoSql. Having an appropriately planned/segmented performance testing approach executed early, and subject to ongoing reporting is critical when deploying systems in a digital transformation.

For the full session recording, click [here](#).



## Brian Brumfield

### The Emergence of Thick-Client Websites: How to Identify and Performance Test Them?

[Brian Brumfield](#) is Director of Performance Engineering & Managed Services at Foulk Consulting. He specializes in APM and performance engineering solutions at Mercury. He also works with ATS/VIP, J9 Technologies, SOASTA, Keynote, Dynatrace, and Foulk Consulting.

On the whole, we're seeing more thick-client development approaches to designing on the web. Today, a fair number of websites behave like mobile apps. As such, they present many challenges to traditional performance engineering practices. So, how do you identify and tackle those sites?

In the past, many developers would shoehorn full-blown websites into mobile frameworks like Cordova, Sencha, etc. Now, we're seeing developers take mobile application architectures and shift them to the web, placing that application architecture within the browser. The birth of a "thick" web client.

As Brian stated during his PAC presentation, "No longer are developers afraid of putting content into the browser. They're loading the heck out of the browser, in megabytes." This challenge presents a fundamental question that Brian poses, "How do you test these things?"

Brian reinforced that these new, thick client websites present unique challenges, that the need for low latency API access is paramount. Also, there's a lot more client-side JavaScript handling business logic outputting rendering activity. New performance testing techniques are required to ensure the responsiveness and timely accuracy of these new types of thick websites.

According to Brian, the essential consideration for performance testing is to not take the millisecond for granted. When it comes to distributed applications, latency can be a significant obstacle. Performance testers need to be aware of how time is used throughout a particular interaction from browser side activity to server-side processing. To strip out wasted time is essential.

Brian demonstrated how to use a website speed and performance optimization service such as GTmetrix to gather information about application performance. He pointed out the poor performing sites are not unique. Particularly so, because many commercial applications will load full-blown business systems into the browser. This incurs significant overhead and creates an impact when it comes to performance testing the application. He recalled situations in which he expected to be able to create a thousand virtual users to use in test cases but was reduced to using only seventy because of the overhead the client-side JavaScript imposed.

Thick web clients also pose a problem due to excessive processing on the client-side that is hard to detect. In the past, a typical web page might make a detectable callback to a server whenever the user clicked a button. A thick web client application can incur processing that hogs client-side resources such as CPU utilization that affects performance. These types of performance bottlenecks are undetectable using traditional request/response monitoring tools.

Brian recommends that you should take a taxonomic approach to performance testing in which all analysis is considered "from a pure workload perspective, from the client to the server." Therefore, the burden is placed on the test designer to gather and organize all the data relevant to the interaction between user and application. This means going beyond measuring only request/response times. Not only do testers need to be able to report throughput regarding where data is coming from, for example, edge-delivered vs. server-delivered, but also examining the burden that the application imposes on the client-side computer hosting the browser. Different tools will be required to create a full understanding of performance overall. It's up to the test designer to make this happen regarding the taxonomy at hand.

He also points out that testing tools need to modernize to provide more comprehensive testing capabilities. Scripts need to go beyond single-use scenarios. Reusability is required to keep costs down while also meeting ongoing testing demands. As Brian says, "approaching these (issues from), a taxonomic workload perspective is probably the best way. It's just going to have to be (this way) until we have scalable, headless browser based testing en masse where we can run thousands of virtual users. It's just a reality today. It will be. Somebody is going to solve that problem. Hopefully, it's Neotys."

For the full session recording, click [here](#).



## Todd DeCapua

### Telemetry: The Essential Ingredient to Success with Agile, DevOps and SRE

[Todd DeCapua](#) is a technology evangelist, passionate software executive, and business leader. Some of his roles/titles include: Executive Director of Global Technology at [JPMorgan Chase & Co.](#), Senior Director of Technology and Product Innovation at [CSC](#), Chief Technology Evangelist at [Hewlett-Packard Enterprise](#), Co-Founder of [TechBeacon.com](#), VP of Innovation and Strategy Board of Director at [Vivit Worldwide](#), and Independent Board of Director at [Apposite Technologies](#). He's also an active [online author and contributor](#) of the [O'Reilly](#) published book titled [Effective Performance Engineering](#).

According to Todd, as companies continue to focus on transformation to Agile, DevOps and Site Reliability Engineering (SRE) sensibilities, many teams are challenged to achieve success. To meet the challenges, they should use telemetry to enable the organization to deliver speed, quality, and automation targets required for successful results. According to him, "Telemetry becomes the key, backbone, and fundamental element that we all need to see."

In his PAC presentation, Todd pointed out that four problems apply for telemetry:

1. Increased speed
2. Meantime to Repair (MTR) reduction
3. Increased collaboration across the organization
4. Delivery of the highest value to the customer

To Todd, a telemetry system is composed of three components: collectors, data and visualization. A collector is the means by which data and information are gathered. A collector might be an application performance or a security monitoring tool.

Data is the information gathered by the collector. He pointed out that given the volume of data that a collector can gather, companies need to have a strategy for collecting the data and for a way to understand and structure the data being gathered.

Visualization is the means by which users interact with the data gathered by the collector. Determining visualization is key to providing value to users and customers. An example of a determining a visualization strategy is to design and implement a dashboard that captures the various data points of interest to the user.

Todd's presentation explores the five types of collectors that engineers should use throughout application development, architecture engineering, and production management to meet the goals and objectives that modern, performance-oriented businesses demand telemetry systems.

- Performance
- Capacity
- Monitoring
- Production Incident
- Security

Todd mentions that the use of streaming collectors is growing in the telemetry landscape. Also, he points out that more telemetry systems are incorporating machine learning, artificial intelligence and predictive analytics in their real-time operation.

Todd emphasizes that telemetric systems must be designed according to user preference. As Todd says, "Is this [dashboard] what your customer and business is after?" Todd recommends taking the time to understand how alerts should be generated as well understanding how information needs to be displayed in a given dashboard.

## Todd DeCapua

When considering the nature and application of telemetry to mission-critical performance testing, Todd offers this wisdom, “As we think about where does telemetry fit? Where does it start? Where does it end? It is throughout. It’s all of these elements (plan, code, build, test, release, deploy, operate, measure). We’ve all seen the individual tools. We know what the data is that comes out of (them). How is it that you can instrument every one of those capabilities, providing that data (in) near real-time so that you can do different types of correlations (to) end up with telemetry as results to enable your customers and your business to drive success?”

Todd recommends reading the following books to learn more about DevOps overall and telemetry in particular.

- [Agile Software Development with Scrum](#)
- [The Phoenix Project](#)
- [Site Reliability Engineering](#)

For the full session recording, click [here](#). You can also find [Todd De Capua’s](#) blog on the topic [here](#).



## Gayatree Nalwadad

### Life Shift in Performance Testing

[Gayatree Nalwadad](#) has ten years of experience as a Principal Quality Assurance Engineer at Fidelity Investments. She’s responsible for all performance testing strategy and execution for her business unit. Before Fidelity Investments, Gayatree worked at Infosys Technology Ltd (India) as a Performance Test Engineer, serving clients such as BofA and Adidas. She has also completed the LoadRunner Certification and Software Testing Certification from SQE Training (a TechWell Company).

In her PAC presentation, Gayatree described her approach to performance testing in an Agile world. She also delved into pain points associated with the Waterfall to Agile transition.

Her presentation’s message assertion - the key to effective performance engineering is always to remain intune with the needs and priorities of users/company executives. She described the six stages of proactive performance engineering: (1) Gathering Non-Functional Requirements, (2) Design, (3) Development, (4) Test Execution, (5) Analysis, and (6) Reporting.

Gayatree feels strongly that all members of the development team are adequately educated and trained on the fundamentals of performance testing concepts, practices, and terminology. She also makes it her mission to make sure performance engineers adjust their test techniques to the development process being used.

When it comes to performance testing in an Agile environment, Gayatree likes to adjust the stages of proactive performance testing throughout the sprint execution model. Non-functional requirements and workload design take place during sprint planning. Test data set-up, writing performance scripts, and Development Independent Testing (DIT) take place amid the sprint. Performance, capacity, stress, spike, longevity and exit testing takes place at the post-sprint retrospective. Gathering application monitoring and capacity data are performed at deployment time. The data collected will be subject to further performance analysis.

## Gayatree Nalwadad

Gayatree also talked about how to conduct troubleshooting, bottleneck identification, citing that she typically makes sure that tools she's using perform as expected. It's not unlike her to inspect the performance scripts closely. She then concentrates on the data. Finally, ensuring that the application is operational, the servers and database are each behaving.

To Gayatree, a sound performance engineer wears many hats, from troubleshooting investigator to system engineer, it depends upon what stage they're focused on during the sprint.

For the full session recording, click [here](#). You can also find [Gayatree Nalwadad's](#) blog on the topic [here](#).



## Alexander Podelko

### Continuous Performance Testing: Myths and Realities

[Alex Podelko](#) has specialized in performance since 1997, specifically, as a performance engineer and architect for several companies. Currently, he is a Consulting Member of Technical Staff at Oracle, responsible for performance testing and optimization of Enterprise Performance Management and Business Intelligence (Hyperion) products.

Podelko's collection of performance-related links and documents (including recent articles and presentations) can be found [here](#); his [blog](#), and Twitter as [@apodelko](#). He also serves as a director for the Computer Measurement Group (CMG), an organization of performance and capacity planning professionals.

If you ask, Alexander will tell you that when it comes to performance testing, "context is king!" He'll probably also recall yesterday vividly...

The old method of conducting load performance testing are typically Waterfall-centric. Performance testing, if only for the enterprise because of the expense, was completed just before deployment. Test activity recording was done low level, typically at the protocol. The test environment was more of a laboratory than an actual production setting, and real testing was to support the checkbox on a requirements document or part of an SLA defining application success.

Just as quick as he remembers how things were, Alexander will reinforce that, "we're not there anymore! Now, we have [many] different types of performance tests in play, and not only as the last step pre-production."

During his PAC discussion, he pointed out what he feels is an apparent paradox. Although the scope and practice of performance testing has improved under Agile and the CI/CD process, the same problem exists. When it comes to choosing between testing for functionality or performance, functionality is given priority. This attitude needs to change.

Alexander then described the current state of performance testing in Agile and CI/CD environments. He talked about the things which need to be considered when incorporating automated testing – "Typically, APIs are more stable which lend themselves well to automation."... His discussion proceeded to evaluate the value of automation relative to cost, summarizing that "full automation" is not feasible nor recommended. Instead, to land on a hybrid balance of manual vs. automated testing based on need.

For the full session recording, click [here](#). You can also find [Alexander Podelko's](#) blog on the topic [here](#).



## Mark Tomlinson

### Cognitive Biases in Performance, Wut?

[Mark Tomlinson](#) is currently a Performance Architect at-large and is also the producer of the popular performance podcast [PerfBytes](#). His testing career began in 1992 with a comprehensive two-year test for a life-critical transportation system - a project which captured his interest in software testing, quality assurance, and test automation. Mark has worked at Microsoft, Hewlett-Packard, and PayPal where he's amassed broad experiences with real-world scenario testing of large, complex systems. Mark is highly-regarded as a thought leader in software testing automation with specific expertise in performance.

According to Mark, there's more to performance engineering than is commonly accepted. As he states, "For most of today's practices in performance engineering, the focus is on the objective measures for the system under test: response times, CPU, disk, memory, network, queue depth. We collect the data and crunch the numbers. We project future capacity. We sleep well. But, what if the objective measures are potentially limiting our understanding of performance? What if those elaborate algorithms had convinced us that we were valuable or right? What if there's more to the story?"

The focus of Mark's PAC content was to tell this untold story of cognitive bias in performance testing and the impact it has on one's ability to expand performance beyond the metrics under consideration.

Mark describes the four Quadrants of the Cognitive Bias Codex:

What should we remember | Too much information

-----+-----

Need to act fast | Not enough meaning

Mark described some types of cognitive biases as they relate to the four Quadrants of Cognitive Bias Codex and performance testing.

- **The fading effect:** We've made great process and methods. We've had no issues for several years. Then we wonder, should we remove this? It's not useful
- **Modality effect:** We only react when we see something turn red
- **Duration neglect:** As issues happen frequently, we start to become accustomed to being a familiar behavior, so we don't pay attention
- **Distinction bias:** As we compare metrics over time, the frequent occurrences (overlays) become prominent in our mind
- **Selective perception:** We pay attention only to things we care about, e.g., slow response time
- **Bikeshedding:** Making a big deal out of a small item to justify our work
- **Money illusion:** The cost of a malfunction is more expensive than we think
- **Inverse money illusion:** Applying an irrelevant technology or technique to a situation implementation cost is cheap
- **Cause of knowledge:** We assume that everyone understands ideas and language that are common to us (geek language)

Another topic Mark touched upon was Cynefin. According to Wikipedia, Cynefin offers five decision-making contexts or "domains": obvious (known until 2014 as simple), complicated, complex, chaotic, and disorder – that help managers to identify how they perceive situations and make sense of their own and other people's behavior. The framework draws on research into systems theory, complexity theory, network theory, and learning theories.

## Mark Tomlinson

Mark distilled the descriptions of the decision-making domains further:

- **Obvious:** tightly constrained, “best practice.”
- **Complicated:** tightly coupled, “good practice.”
- **Complex:** loosely coupled, “emerging practice.”
- **Chaotic:** lacking constraint, “novel, untested practice.”
- **Disorder:** none of the above

The benefit of Cynefin is that decision making based on risks helps you to determine the right decision. Testing strategy and applying solutions are best considered concerning Cynefin.

Overall, performance testers need to decide what needs to be remembered, what is important and how to avoid an overload of information. All of these considerations are the subjects of cognitive bias.

For the full session recording, click [here](#).



## Zak Cole

### Blockchain Performance Engineering

[Zak Cole](#) is CTO of Whiteblock, a performance engineering company focused on blockchain and distributed ledger technologies. At Whiteblock, Zak developed a blockchain testing platform, the first of its kind, to help development teams build compelling and scalable blockchain systems. As an early investor in Bitcoin and Ethereum, he has been involved in the blockchain space since 2013 even serving as a network engineer in the United States Marine Corps. Additionally, he contracted as a front-end developer with Google and worked as an information systems security professional and technical product manager with Apposite Technologies.

While the blockchain space is still very much in its infancy, the promise of distributed ledger technology presents valuable and unique solutions for a variety of global issues. However, to ensure true mainstream adoption, the blockchain development community is racing to overcome several key performance issues which bottleneck these adoption efforts.

During his PAC presentation, Zak summarized the performance issues (and solutions) related to blockchain technology. His presentation was a call to action for the performance community to make contributions, especially those with enterprise experience.

Attendees were also exposed to some results associated with a series of tests that Cole recently conducted at Whiteblock, a brief overview of the methodologies applied.

Cole concluded with a review of the technology behind blockchain, discussing specific products like [Hyperledger](#), [Ethereum](#), and [Plasma](#). He described how blockchain is used to support cryptocurrency, diving into the dynamics of bitcoin mining under blockchain, interoperability, and blockchain security.

For the full session recording, click [here](#).



## Wilson Mar

### Lessons Learned Installing Client MacOS

[Wilson Mar](#) is a Principal DevOps Performance Evangelist at JetBloom where he focuses on being proactive with performance/capacity issues while implementing complicated software in enterprises and entrepreneurial organizations. His passion is getting large complex secure enterprise systems to work well – a difficult challenge that he strives to meet daily.

Over the years, Wilson has developed a technique that allows him to install the many applications he typically uses on any laptop. He's applied this technique to help others developers maintain their computers as well.

The focus of Wilson's PAC presentation was to share the lessons he learned when attempting to implement effective, time-saving, developer-focused installation scripts for OS X clients. On one of his slides, he quoted [Mary Meeker](#) at Kleiner Perkins (about technology adoption), "... it took 75 years for dishwashers to be adopted as a general technology, while consumer access to the internet took less than ten years."

Wilson is a proponent of leveraging [bash](#) as a goto tool for client-side software installation. Mastering bash is an essential skill for those who want to support the effective delivery of development tools to teams in the easiest way possible.

Wilson left his audience with some protips:

- Automated tuning is now an expected feature, particularly in cloud deployments.
- We need to become data scientists to understand more about everyday computing and statistics.
- We should forget capacity. It's dynamic and infinite.
- There is still a need for performance testing in the serverless environment.
- When doing sprint planning, include time to upgrade tooling before each iteration.
- Include time in sprint planning to agree on and establish a team toolset.
- As obvious as it seems, all installation scripts should use variables and configuration schemes, not hardcoded values.
- When working with data script downloaded from a particular cloud environment, execute them from outside the given cloud environment.
- Use [pushd and popd](#) to go temporarily to a directory.
- Utilize package managers such as [Windows Chocolatey](#), [brew for Mac](#), [apt-get](#) for ubuntu/debian, [yum](#) for Red Hat and Centos, and [zypper](#) for Suse.
- Lint the Bash and Python scripts.
- Beta test your scripts before releasing them to a broad audience.

The benefit of using scripts to maintain software on client machines is a better use of time in the workday. As Wilson says, "I spent most of my time debugging my thinking rather than debugging my typos."

For the full session recording, click [here](#).





## Jonathon Wright

### Digital Performance Lifecycle: Cognitive Learning (AIOPS)

[Jonathon Wright](#) is a digital therapist, strategic thought leader, and distinguished technology evangelist. He specializes in emerging technologies, innovation and automation, and has 18+ years of international commercial experience within global organizations. He is currently the CTO of Digital-Assured in Oxford, UK.

During his session, Wright defined Digital Experience (DX) and its importance within the Digital Performance Lifecycle (DPL) as it helps organizations adopt Minimum Viable Performance (MVPx), Performance as Code (PaC) combined with Test Data as Code (TDaC) to supercharge the Continuous Performance (CPx) delivery pipeline using bots (ChatOps) to deliver higher quality software at an ever-greater pace. According to him, digital business systems need to be extremely responsive to change and resilient in the face of unpredictable failure modes across machine to machine (M2M) and Internet of Things (IoT) connections and microservice endpoints.

Ultimately, Wright recommended to his audience that they start arming themselves with the knowledge and understanding to unlock the Cognitive Knowledge gap to create value-driven delivery. Stating further that becoming insight-driven through the cognitive adoption of digital capabilities and technologies is a must.

For the full session recording, click [here](#). You can also find Jonathon Wright's blog on the topic [here](#).



## Tingting Zong

### Integrate Performance Test with DevOps

Tingting Zong graduated from Zhejiang University with a Master's in Computer Science. She currently leads a team of 20 performance engineers at Insigma Hengtian Software Ltd.

With more than eight years of performance engineering experience behind her, Zong has successfully overseen more than 200 software performance testing projects.

During her session, Zong walked the audience through how to best integrate multiple performance tests as individual DevOps steps associated with many tasks. Her main message - the criticality of a standardized software delivery process. She reinforced the idea that continuous integration and continuous testing are requirements which help pave the way to rapid submission, feedback, and efficiency.

For the full session recording, click [here](#).

## Putting it All Together

The June 2018 gathering of the Neotys Performance Advisory Council presented information that is a must-read for any professional test practitioner. Presentations spanned an array of topics that focused on tools/techniques relevant to modern performance testing and the thinking by which test practitioners can expand and enhance their approach. Performance testing continues to be a critical aspect of the software development lifecycle, and we want to make sure we continue to provide the venue and the platform for sharing and learning.

The information and wisdom shared during PAC 2018 will go a long way toward improving the way software is developed worldwide. But more importantly, software that is the result of the new techniques and practices shared during these discussions will enhance the lives of creators and users everywhere.

