



Test Automation in Context: What Might Help?

Let's talk and think about when, where, why, and how to effectively apply automation as part of a complete testing approach or strategy. We're looking for practitioner experience and constructive advice to help people succeed. Of course automation doesn't replace thinking testers – but how might it support and aid broader testing efforts?

How could you describe the automated checks and exploration that have been helpful? Which aspects of your context had the greatest effect on the success (or failure!) of your efforts? Which hopes and expectations are reasonable to have and achievable for test automation? What lessons have you learned from attempting to use tools to support your testing?

Selected Talks will be repeated for an online audience shortly after CAST.

Monday Tutorial



Tariq King

Testing AI and Machine Learning: A Holistic Strategy

Although there are several controversies and misunderstandings surrounding AI and machine learning (ML), one thing is apparent — people have quality concerns about the safety, reliability, and trustworthiness of these types of systems. Testing ML is challenging and requires a cross-section of skills, and experience from areas such as mathematics, data science, software engineering, cyber-security, and operations. In this tutorial, Tariq King introduces you to a holistic strategy for testing AI and machine learning systems. You'll start with AI/ML fundamentals and then dive into approaches for testing these types of systems offline, prior to release, and then online, post-deployment. Engage with other participants to develop and execute a test plan for a live ML-based recommendation system, and experience the practical issues around testing AI first-hand. By the end of the session, you'll be able to better prepared to help your organization build trustworthy machine learning systems.

Biography

Tariq King enjoys breaking things and is currently the Chief Scientist at test.ai, where he leads research and development of their core platform for AI-driven testing. He started his career in academia as a tenure-track professor and later transitioned into the software industry. Tariq previously held positions as a test architect, manager, director, and head of quality. He has published over 40 articles in peer-reviewed software testing books, journals, conferences, and workshops. He is a member of the ACM, IEEE Computer Society, and Association of Software Testing, and serves as a board member and keynote speaker for several international software engineering and testing conferences.

Tuesday Plenary Sessions



Ben Simo

Computer-Assisted Testing

An experience report.

People often classify testing as being either manual or automated. Some even label testers (including themselves) as either manual or automated. This is a peculiar distinction, and a false dichotomy.

Programmers don't describe themselves or their work as manual programming or automated programming. Programmers use automation, and they use their hands; but most importantly: they use their minds. They use their minds to analyze human problems and develop software solutions to those problems. In the same way, software testers use their minds to analyze problems and solutions in order to design experiments that demonstrate how software works and uncover problems. Testing cannot be automated any more than programming can be automated.

Although testing cannot be automated, automation is essential in testing. If your test design, execution, or analysis of results includes a computer, it is supported by automation — it is computer-assisted testing.

Join Ben Simo as he chronicles his real-world experience practicing Computer-Assisted Testing using a variety of techniques throughout the past 30 years. Hear how Ben has used automation in test design, test execution, analysis of results, and communicating what matters. Learn about methods for using automation to do much more than push buttons and watch screens. Learn to persuade your colleagues to build tools that improve your testing. This experience report includes the following techniques, and more:

- Interactively guiding testers through procedures and collecting results
- Dynamically generating test cases from state models
- Using relational data models to evaluate data quality
- Building alternative implementations to serve as oracles
- Driving load tests with analytics data
- Testing by observing production systems

Biography

[Ben Simo](#) is a software quality leader who has been helping make software better for over 30 years. Ben has applied his skills as a software tester, designer of testing tools, and quality coach in a variety of industries, including: healthcare, finance, defense, education, marketing, ecommerce, and cloud services.

Ben approaches software testing as something much larger than verifying that explicit expectations are met. He believes that good testing includes observational and experimental investigation that enables people to make better decisions that lead to delivering better software. Although Ben believes that testing cannot be automated, automation plays an essential role in most of his testing work.

Ben is a member of the [Context-Driven](#) school of testing and a former president of the [Association for Software Testing](#).

Ben is perhaps [best known](#) as the “[sort of skilled hacker](#)” who uncovered [numerous problems](#) with the initial release of the US Government’s health insurance marketplace: Healthcare.gov.

Ben is currently exploring and creating tomorrow’s testing tools as a product researcher and product manager at [Tricentis](#).



Rajni Hatti

Ethical Hacking for Testers

Security testing is often handled by a specialized team or a set of automated tools, but every tester should understand the basics of how malicious data can enter a system in order to prevent the vulnerabilities from occurring in the first place. In this session, I will share a case study from a healthcare technology project where I led the QA initiative. I will show you the ethical hacking practices I used to find security flaws in various contexts such as design, code review, testing, and product release. I will show you the practices I used that worked to:

- Determine the attack vectors most likely to occur based on the application functionality.
- Determine what security tests are needed based on the likely attack vectors.
- Create valuable tests to identify common security flaws.
- Decide if and how a security testing tool can aid testing efforts.

You will learn how to be an ethical hacker in order to make your system more robust, and you will leave with a better understanding of how to incorporate a security mindset into your daily testing efforts.

Biography

Rajni Hatti has a Computer Science degree from Cornell University and has worked as a software professional for over 20 years in a variety of industries, including Telecom, Finance, Healthcare Technology, and Online Fraud Detection. She began her career as a software developer and gained a passion for testing, which led her to using her technical skills as an automation engineer, and her leadership skills as a manager and team coach. She then went on to start her own software consulting company, successfully driving projects from conception to deployment. Rajni currently works as a Lead Software Test Engineer for MaxMind, an industry-leading provider of IP intelligence and online fraud detection tools.



Greg Sypolt

Building a Better Tomorrow with Model-Based Testing

Let's build a better tomorrow and a more equitable world.

We have all heard the call for change, and EVERFI is committed to answering it with educational courses and quality digital transformation with model-based testing. In a world of agile development, we experience a fast-paced development environment that is challenging, if not impossible, to keep up with hundreds of courses that require testing across our platforms. It calls for change, and EVERFI is committed to answering it by introducing model-based testing.

It's a software testing technique that helps you simplify and accelerate application development without jeopardizing quality. By shifting to model-based testing techniques, the system is broken down into smaller manageable components. The models describe the system requirements and capture the expected behavior. Then utilizing test generation algorithms will recreate the entire system from a collection of models to generate Cypress test code from those visual flowchart models.

Biography

As a VP of Quality Assurance at EverFi, Greg Sypolt enjoys speaking and writing about how others can rethink how they approach quality excellence. He has spent most of this career in quality assurance with an engineering mindset, gaining experience in quality tool development, automated testing, and DevOps. Being a quality advocate, he believes delivering high-quality products is everyone's responsibility.



Tariq King

Towards Better Software: How Testers Can Revolutionize AI and Machine Learning

You may have heard that software ate the world and AI is eating software. However, if you've been paying attention, then you've probably realized that the world is filled with bad software. Many organizations struggle with meeting their quality goals and keeping testing-related costs contained. Software is indeed revolutionizing the world, but the world is also paying a revolutionary price for bad software. So where do AI and machine learning (ML) fit in? Are AI/ML breakthroughs just new ways of filling the world with bad software? Or do they offer a path towards better software?

Tariq King believes that although ML is bringing valuable improvements and capabilities to software, it is unlikely for such advances to be successful without integrating testing and quality engineering into AI/ML workstreams. Join Tariq as he highlights the parallels and differences between ML engineering and software testing, and explains why he is convinced that testing professionals hold the keys for the transition to AI/ML system development. However, before testers can revolutionize AI/ML, they must first become agents of change through innovation. For the first time, Tariq will share his own journey and transformation, and discuss how he is helping other individuals, teams, and organizations on the road to AI.

Biography

Tariq King enjoys breaking things and is currently the Chief Scientist at test.ai, where he leads research and development of their core platform for AI-driven testing. He started his career in academia as a tenure-track professor and later transitioned into the software industry. Tariq previously held positions as a test architect, manager, director, and head of quality. He has published over 40 articles in peer-reviewed software testing books, journals, conferences, and workshops. He is a member of the ACM, IEEE Computer Society, and Association of Software Testing, and serves as a board member and keynote speaker for several international software engineering and testing conferences.



Laurie Sirois

Quality Isn't Funny

How can software testing professionals help their organizations take Quality even more seriously? Many of the challenges testers face are consistent across industries. This humorous, memoir-style talk will bring light to such serious topics as: elevating quality professionals to 1st class citizen status; getting buy-in on “Shift-Left” testing practices & true continuous improvement culture; how to add value by identifying risk, representing the customer, honing requirements and planning testing efforts earlier in business discussions; and how to sustain progress by dropping enough of our defenses to take ownership to the next level (organizational).

Humor doesn't always have a place in preventing contentious escalations, but having data to back your recommendations & decisions is only one step in gaining support. Data doesn't tell stories, people do: humor during stressful times can build trust and the strong relationships needed to get serious issues addressed sooner. Similarly, data doesn't make decisions, people do. Leveraging self-awareness and emotional intelligence tools can help us reach consensus sooner & underscore the seriousness of our daily profession without taking the fun out of our careers.

“The Life You Save May Be Your Own” ~Flannery O'Connor

Biography

Laurie Sirois is currently the Director of Quality Assurance at PTC Inc. & has over 25 years of experience in QA leadership roles, as well as Product Owner and Producer roles at small, mid-sized and global companies. She specializes in building and evolving QA teams, processes and automation initiatives that close the gap between business and customer needs.

Laurie holds an active CMSQ (Certified Manager of Software Quality – QAI) as the top-scoring participant globally as of 2015. She has also earned her CPCU (Chartered Property Casualty Underwriter – The Institutes), and holds active CSPO and CSM designations from the Scrum Alliance. She champions Quality as a core component of enterprise risk management, and her favorite motto is “fail fast, succeed faster.”



Angie Jones

The Build That Cried Broken

There's a famous Aesop Fable titled "The Boy Who Cried Wolf". As the story goes, a young shepherd-boy would declare that a wolf was coming in an effort to alarm the villagers who were concerned for their sheep. The boy got a reaction from the villagers the first three or four times he did this, but the villagers eventually became hip to his game and disregarded his future alarms. One day, a wolf really was coming and when the boy tried to alert the villagers, none of them paid him any attention. The sheep, of course, perished.

For many teams, their continuous integration builds have become just like this young shepherd-boy. They are crying "Broken! Broken!" and in a state of panic, team members assess the build. Yet, time and time again, they find that the application is working but that the tests are faulty and giving false alarms. Eventually, no one pays attention to the alerts anymore and have lost faith in what was supposed to be a very important indicator.

Let me help you save the sheep...or in your case, the quality of your application. In this talk, I'll share a personal story of how our builds were a broken mess – littered with hundreds of untrustworthy tests. I'll share with you the step by step process of how we prioritized this issue while still managing to continue feature development, and how we ultimately returned our builds to a healthy state.

Key Takeaways:

- How to build stability within your continuous integration tests
- Tips for managing tests that are failing with just cause
- How to influence the perception of the credibility of the tests among stakeholders

Biography

Angie Jones is a Java Champion and Senior Director who specializes in test automation strategies and techniques. She shares her wealth of knowledge by speaking and teaching at software conferences all over the world, as well as and leading the online learning platform, Test Automation University. As a Master Inventor, Angie is known for her innovative and out-of-the-box thinking style which has resulted in more than 25 patented inventions in the US and China. In her spare time, Angie volunteers with Black Girls Code to teach coding workshops to young girls in an effort to attract more women and minorities to tech.