Association for Software Testing

CAST 2006: Influencing the Practice
June 5th-7th, 2006 – Indianapolis
http://www.associationforsoftwaretesting.com/conference/index.html

Event Overview

The first annual Conference of the Association for Software Testing (CAST) will be 3 days of interactive learning and discussion on the theme of "Influencing the Practice", in The Crowne Plaza at historic Union Station, Indianapolis, Indiana.

What makes CAST unique?

What distinguishes this conference, is that ALL ideas are open to challenge, debate, and discussion. The conference will allow a high level of interaction among presenters and attendees, and flexibility will be built into the schedule so that the energy for a particular topic can modify the course of conference events.

At 15 minutes before the end of each talk, Q&A will begin and active attendee participation will be invited. If a lively discussion needs more time, the topic can be adjourned to another space to keep the conversation going.

Being a small conference, the chances to confer are great. Please take full advantage of this unique, exciting opportunity and join us at CAST.

Keynotes

Exploratory Testing After 23 Years - Cem Kaner

Twenty-three years ago, I coined the phrase "exploratory testing." I didn't invent the practice but, as far as I can tell, I was the first public advocate of it. The idea was disparaged widely, sometimes thoughtfully. Many of the discussions have been and still are whether it is a good / bad idea in principle, rather than how to do it more effectively and how to assess the quality of the work done. Twenty-three years later, some of the same old attacks on the idea are still widely repeated. Why? What legitimate problems and concerns are they addressing? To what extent has our progress as explorers addressed those concerns? What problems have we actually solved and what should we be telling test groups about how to develop effective exploratory skills and practices in their work? Several well known explorers are at this conference. This talk opens a discussion that I hope to see as a theme throughout the meeting. It's time to publicly take stock, to identify areas of agreement and areas of controversy, areas of progress and areas of ongoing concern.

Against Certification - James Bach

Our profession is maturing - or at least it's getting older - and some have suggested that it's time that we, as a community, establish a process of certifying the quality of testers. Some people have acted on this conviction, either by establishing a certification program or by getting certified themselves. I think the programs that currently exist are poorly founded, financially motivated, and are actually an embarrassment to our vocation. In this talk I will lay out the issues as I see them, and also talk about what a good certification program will look like when someday our craft actually is mature enough to do certification right.

	Room / Location		
Day 1: Tutorials	**Victoria Station A**	**Victoria Station B**	**Victoria Station C**
8:00 AM – 9:00AM	Continental Breakfast		
9:00 AM -10:30AM	**Cem Kaner:** *Creating Your Own In-House Testing Course*	**James Bach:** *Self-Education for Testers*	**Hung Nguyen:** *Strategies and Tactics for Global Test Automation*
10:30 AM– 11:00 AM	Break		
11:00 AM - Noon	Tutorial continues	Tutorial continues	Tutorial continues
Noon - 1:00 PM	Lunch in **Grand Central Station ABC**		
1:00 PM - 2:30 PM	Tutorial continues	Tutorial continues	Tutorial continues
3:00 PM - 3:30 PM	Break		
3:30 PM - 5:00 PM	Tutorial continues	Tutorial continues	**Julian Harty:** *Six Thinking Hats for Software Testing*
5:00 PM - 6:00 PM			

	Room / Location			
Day 2: Conference	**Victoria Station**	**Victoria Station AB**	**Victoria Station CD**	**Pennsylvania Station**
8:00 – 9:00 AM	Continental Breakfast			
9:00 – 10:25 AM	**Cem Kaner:** Keynote - *Exploratory Testing After 23 Years*			Exhibits
10:25 –10:40 AM	**Pat Schroeder:** *Competition Announcement*			
10:40 – 11:00 AM	Break			
11:00 AM - Noon		**Douglas Hoffman vs. James Bach:** *DEBATE: How Early Should We Begin Testing?*	**Andy Tinkham:** *Lessons Learned in Teaching Test-Driven Development*	
Noon - 1:00 PM	Lunch in **Grand Central Station ABC**			
1:00 – 3:00 PM		**Bernie Berger:** *An Overview of Software Testing in the Financial Services Industry*	**Chris Johnson:** *Operational Mission Thread Testing*	
			Iain Mackenzie: *Beyond Performance Testing: Performance Assurance – A Full Product Life Cycle Discipline*	
		Jorge Correa de Oliveira, Cidinha Costa Gouveia, & Romulo Quidute Filho: *A Way of Improving Test Automation Cost-Effectiveness*	**Allen Johnson:** *Rainbow Nets to Promote Understanding of What Needs to be Tested*	
			Martin Taylor: *EPDAV – A Model for Test Case Definition*	
3:00 - 3:30 PM	Break			
3:30 – 5:00 PM		**Julian Harty:** *Security Testing Using JUnit and Perl Scripts*	**Scott Barber:** *Effects of Modeling Software Application Usage Visually and Intuitively*	
5:00 – 7:30 PM	Break			
7:30 – 9:00 PM	**Robert Sabourin & Ross Collard:** *A System Performance Testing Study Guide*			

	Room / Location			
Day 3: Conference	**Victoria Station**	**Victoria Station AB**	**Victoria Station CD**	**Pennsylvania Station**
8:00 – 9:00AM	Continental Breakfast			
9:00– 10:30 AM	**James Bach:** Keynote - *Against Certification*			Exhibits
10:30– 11:00 AM	Break			
11:00 AM - Noon		**Karen Johnson:** *Working Through Language, Time, and Cultural Differences*	**Douglas Hoffman:** *The Darker Side of Metrics*	
			Kenn Petty: *Transitioning From Standard V&V to Rapid Testing Practices in a Chaotic Project Environment*	
Noon - 1:00 PM	Lunch in **Grand Central Station ABC**			
1:00 – 2:40 PM		**Peter Zimmerer:** *Insights in Real Test-Driven Development*	**Bernie Berger:** *Generating Test Cases with All-Pairs*	
			David Gilbert: *Exploratory Testing - The Art of Exploration*	
		Stuart Reid: *The Personal Test Maturity Matrix*	**Vandana Shah:** *Effectively Managing Testing Efforts on Legacy and New Applications*	
2:40 - 3:00 PM	Break			
3:00 - 3:30 PM	**Pat Schroeder:** *Competition Results*			
3:30 – 5:00 PM	AST Annual Meeting			

Tutorials

Creating Your Own In-House Testing Course - Cem Kaner

Short courses can provide valuable overviews, but it is hard to structure their time to develop skills or shared insights. There is too much information at once and too little time for discussion, practice, and private reflection. For the last five years, I've been developing software testing courseware that workgroups can download for free, adapt, and work through at their own pace. You can see much of my progress at http://testingeducation.org/BBST. This year, I've started using a free course management program (Moodle) that makes the structuring of the course easier and provides support for student interaction out of the classroom. This tutorial will introduce you to Moodle, to the instructional style, to my collection of course materials and other useful free content on the web, and some lessons that I've learned (and references that I use) as I continue to develop skills (not all there yet) in creating and managing learning activities.

Participants MUST bring their own laptop and it must be able to make a wireless network connection.

Self-Education For Testers - James Bach

No budget for classes? Not enough time to get an advanced degree while earning a living? I'm working on a book about self-education. It's about how to learn anything you want when you are on your own. In this tutorial I will talk about self-education specifically for testers. You are probably already a self-educated tester, but how systematic is your education? How thorough is it? In addition to talking about self-education in general, I will lay out some of topics of suggested study, and illustrate how my eclectic education has been a competitive advantage for me in my consulting practice.

Strategies and Tactics for Global Test Automation - Hung Nguyen

We automate software testing to gain speed. We organize our distributed teams globally to maximize around-the-clock coverage and cost efficiency. Both solutions fulfill legitimate objectives. However, implementing them successfully while keeping the risks contained with a high degree of certainty proves to be an enormous challenge. Regardless, development and testing teams are asked to do more and to align testing effort with the business objective. In that, we are asked to deliver a holistic transformation that brings the best out of test automation and distributed/global resources. In this tutorial, we will discuss seven specific steps that will deliver high and visible return through a global test automation program:

* Assess testing strategy and needs
* Leverage automation technology to maximize speed
* Minimize costs and risks introduced by global resources
* Use the right mix of test automation technology for the job
* Align testing with business process and development practice
* Secure and develop competent resources
* Measure, analyze, and optimize for continuing improvement

You will learn through technical and management case studies and real-world examples. In a nutshell, this tutorial discusses the training for people on both sides of the continents, automation technology and implementation, process, and alignment needs with business strategy.

Six Thinking Hats for Software Testing - Julian Harty

Software still has bugs, lots of them, and even though software testing is now recognized as one way to help identify bugs, the testing is seldom satisfactory for anyone involved in the process. This tutorial provides fresh ideas and techniques to help improve the effectiveness and efficiency of software testing by combining some of the best ideas from both inside and outside the testing industry.

Sessions

Day 2

Debate: How Early Should We Begin Testing? - James Bach vs. Doug Hoffman, Moderated by Jon Bach

James Bach and Doug Hoffman have differing views on when testing should begin. Simply stated, James takes the stand that testing should begin as soon as possible and Doug the view that testing should be delayed until the software is complete. The session is structured around their disagreement, with each party presenting and arguing their viewpoint.

After presenting and arguing their points, the discussion will be opened to attendees for questions and comments. The facilitator (Jon Bach) will also be pointing out some of the communication techniques being used to express and elicit conflicting viewpoints - as we'll be doing in all of the CAST sessions.

Lessons Learned in Teaching Test-Driven Development - Andy Tinkham

In this presentation, Andy Tinkham will discuss lessons he has learned in teaching test-driven development (TDD) to undergraduate and graduate students at Florida Tech. The presentation looks at the differences between students at different levels (first-year undergraduates, third- and fourth-year undergraduates, and graduate students) and the differences that arise due to differences in course focus (introduction to programming, unit testing, and a semester-long immersion into a tool). Andy will then attempt to generalize the lessons to situations more germane to practitioners working in the field of software testing.

An Overview of Software Testing in the Financial Services Industry - Bernie Berger

Testing software in context of domain-specific systems presents interesting considerations for testers. One example is the NASDAQ, which in reality is nothing more than computer software running over a very large network. An electronic exchange is special in that there is no physical space where high-strung traders are yelling, frantically waving their arms, and giving hand signals to each other from across the room, where thousands of paper order tickets fall to the floor like confetti, and where, after the closing bell is rung, traders manage their stress at "Harry's" over a few 18-year old single malts. Instead, an electronic exchange is comprised of software running on distributed architected servers, high-speed data feeds, GUIs, middleware, messaging queues, and databases. The business is the technology; the technology is the business.

This paper will present a background of the Financial Services industry, describe different aspects of its software, and discuss some of the problems that face software testers in financial firms.

A Way of Improving Test Automation Cost-Effectiveness - Jorge Correa de Oliveira, Cidinha Costa Gouveia, Romulo Quidute Filho

Test automation has become more and more popular as the market demand for more complex software, involving higher risks and using the same or fewer resources in development, has increased. However, testing automation is uninviting, as the success rate is so low. A number of research papers discuss the problems faced in the test automation process, such as the complexity of automation, poor choice of tools, and the effort spent to automate. This paper proposes a test automation viability analysis method based on a mathematical procedure which, in this team's experience, tends to increase the chances of finding cost-effective test automation processes that achieve real or lasting benefits.

Operational Mission Thread Testing - Chris Johnson

Effective testing is a vital component of every large software development effort. The goal of such testing is ensuring the maximum capability for the end user, while delivering a product that still allows for increased profit and minimized rework. However, the size and complexity of such software efforts often result in a system that is difficult to test in a manner that is consistent with this goal. Testing becomes an exercise in checking small pieces of total functionality, often without the context of the overall system capability. The result is sub-optimized testing that frustrates the customers, as well as the development team.

Operational Mission Thread Testing was developed to address this problem. It has the benefit of producing end user capability faster than traditional testing methods. The methodology has benefited users fighting in battlefields where speed of delivering capability is critical to human life, as well as giving software developers market edge over their nearest competitor. This presentation describes the application of this technique to a specific large and complex system developed for the Department of Defense.

Beyond Performance Testing: Performance Assurance – A Full Product Life Cycle Discipline - Iain Mackenzie:

Performance testing is, by definition, thought of as a testing phase discipline. For enterprise class organizations we advocate a dedicated performance assurance group, whose scope encompasses not only testing but all phases of the product delivery lifecycle from requirements and design through build, deploy and manage.

What is Performance Assurance? It is the integration of performance into all stages of the Product Delivery Life Cycle: requirements, architecture, design, modeling, build, testing and monitoring. It is a combination of skills, process and tools

Using Rainbow Nets to Promote Understanding of What Needs to Be Tested - Allen Johnson

This paper presents an example of Rainbow Nets being used to develop tests for a software tool designed to simulate the movement of cattle. Artifacts created for the design specification using RNs are used to create software tests before,

during, and after the design implementation. The Rainbow Net artifacts facilitate achieving maximum test coverage. Creating tests before the implementation begins allows for rapid verification of the design specification itself and leads to quickly understanding what and how the design specifications should be implemented and tested. This approach dictates an object oriented design. Both the functions to be implemented and the decisions that drive those functions can be partitioned into various RN structures to promote maximum understanding of the design. Furthermore, this approach facilitates incremental design. RN structures can enable the designer or tester to look at the problem/solution in a way that leads to observations that would otherwise be lost.

EPDAV – A Model for Test Case Definition - Martin Taylor

This paper describes a theoretical model for defining test cases, known by its acronym "EPDAV". It shows how the model was derived and how it can be applied to any System Under Test (SUT). Variations on this model are illustrated, with practical examples from Martin Taylor's testing experience at Texas Instruments. The paper also presents some of the challenges selecting or building a Test Management System (TMS) and an Automated Testing Framework (ATF) that supports this model.

Security Testing Using JUnit and Perl Scripts - Julian Harty

This paper describes a recent practical experience where JUnit was used for testing security bugs in addition to functional bugs. Perl scripts were also used during the exploration phase. The application being tested was mature, but insecure. By using a combination of JUnit and Perl scripts it was possible to create effective and efficient automated test cases. A number of supporting tools made the work easier.

Based on his experience of this work Julian Harty suggests a similar approach may be fruitful for testing security of other types of networked applications, including web-based and XML-based software.

Effects of Modeling Software Application Usage Visually and Intuitively - Scott Barber

The effectiveness of performance testing in terms of the accuracy of load simulation is a common topic of conversation and concern among those that build or depend on web-based applications, testing a Web-site in such a way as to reliably predict performance. However it is still often more of an art than a science.

Analysts, developers, testers, managers and executives do not speak the same language. They do not use the same tools or techniques to describe software and no attempt to bridge this gap to date has been widely accepted. UML is too technical and obscure for many. Use Case documentation is often too wordy and lacking in technical detail and flow. A technique derived from a white-board accident in 2001 has taken a life of its own in bridging this communication difficulty, in more than a few household name corporations around the world, because it is intuitive, simple, informative and exactly as formal as the corporation needs it to be. This paper describes this technique, its evolution in industry and the plans to continue evolving this technique in the future.

A System Performance Testing Study Guide - Robert Sabourin and Ross Collard

A System Performance Testing Study Guide for Undergraduate Software Testing Students was developed by Robert Sabourin and Ross Collard based on the Commercial Case Study. The Commercial Case Study is used to support teaching System Performance Testing to mid-level professionals. The Commercial Case Study has been used as a basis of several training programs based on one- though three day tutorials. The study guide developed was used in teaching System Performance Testing as part of the undergraduate Software Engineering course "ECSE 429 – Software Validation" at McGill University. This paper outlines the approach used to develop the study guide, use of the study guide in a semester-long undergraduate software validation course and recommended changes which are expected to be implemented in subsequent undergraduate studies.

Day 3

Working Through Language, Time, and Cultural Differences - Karen Johnson

You might be working on a virtual team with members scattered through multiple countries, or you might be managing outsourced resources in a different country. In Karen Johnson's experience, it has been several years since she has worked on a project that did not include at least one of the following: 1) a virtual team with members located in multiple countries, 2) contracted resources in a different country, and 3) an employee for whom English is a second language (known as ESL). While Karen enjoyed the exposure to different cultures and working in a global community, she has inherited some new work challenges. After reflecting, she has realized that most of these new challenges can be grouped into three areas: differences in language, culture, and time. Her paper and presentation include some insights and guidance on handling these three challenges, both as a co-worker and as a manager.

Transitioning from Standard V&V to Rapid Testing Practices in a Chaotic Project Environment - Kenn Petty

This is a two-year case study of a small ISV's response to a colossal shift in the way they did business. New products needed to be developed, using new technologies, using new methodologies, under extremely tight schedules. The staff doubled with the influx of contractors and all the issues this can cause happened. The requirements management was lax. There was no design documentation. Schedules were pressed with 100% allocation determining schedules. Reliance on documented designs, clear and accurate requirements, and lots of time to test them were not in the cards. What could the QA staff do? Rapid Software Testing with its emphasis on collaboration, interaction and reduced time on documentation was a life saver.

This presentation will cover how the application of the principles of Rapid Software Testing and the Context-Driven School of software testing helped the QA staff succeed in this challenging environment.

Insights in Real Test-Driven Development - Peter Zimmerer

Today, many people talk about test-driven development (TDD) and there is some hype to perform test-driven development in software projects. In this situation, when many people tell you to do TDD, it cannot be bad to ask the question: what is really behind TDD?

In this paper Peter Zimmerer wants to share his view of TDD's advantages and disadvantages and how the TDD concept can be extended to all levels of testing. Based on experiences gained from real-world projects employing TDD, Peter will explain how to use TDD practices that support preventive testing throughout development and result in new ways of cooperation between developers and testers. This helps us to see new aspects of test-driven development and to get a better understanding how it fits into the big picture of software testing and development.

The Personal Test Maturity Matrix - Stuart Reid

This paper introduces a new framework called the Personal Test Maturity Matrix (PTMM), the purpose of which is to provide career guidance for software test practitioners. The PTMM is a new idea, and, as such, has not yet been tested in practice; the purpose of this paper is to gain feedback on the idea.

The main concept behind the PTMM is that testers, especially those new to the discipline, could benefit from using a framework that describes the different roles that software testers perform and the skills required to do this. The PTMM thus comprises roles and skills, which are categorized into four areas: test skills, IT skills, soft skills and domain knowledge. The idea is that an individual tester could identify their own specific skill-space, which describes their capabilities in these four areas, and then use the PTMM to provide them with suggestions on which skills they might wish to acquire next. Job satisfaction and motivation is also explored.

Generating Test Cases with All-Pairs - Bernie Berger

The All-Pairs testing technique generates test cases where every value of every variable is paired with every other variable value. Although the previous sentence sounds confusing at first, it becomes easy to understand with a working example, as explained in this presentation.

Exploratory Testing - The Art of Exploration - David Gilbert

In this presentation, David Gilbert will present his ideas on documentation and results presentation for exploratory and manual testing. The presentation is supported by David's eight years of experience as a consultant in the industry, primarily in support of automated testing. It looks at the differences in results presentation between manual testing and automated testing, and the different perceptions those presentations create as relates to quality and value being derived from either process. David proposes a shift in results presentation from the current standard to one closer to automation, to improve the perception of the value of exploratory and manual testing, especially to those outside the test team, and to help these processes gain wider acceptance as value added methodologies.

Effectively Managing Testing Efforts on Legacy and New Applications - Vandana Shah

Many organizations are choosing to adopt developer testing to produce better quality software and more maintainable applications. However, implementing developer testing can be challenging for organizations with large legacy applications that have no associated unit tests. Yet, with the right process and tools, organizations CAN test entire legacy applications, developed over several years, to build quality and confidence in their software.

As an implementation consultant at Agitar Software, Vandana Shah has spent several years consulting with customers who are adopting unit testing for both legacy code and new applications. Based on her experience, this presentation will highlight the key steps and tools that enable a successful transition to incorporating developer testing practices on legacy code. The presentation describes the transformation of a largely waterfall-based development strategy into an agile, test-driven practice.

Additional Events

- Vendor exhibit space
- Testing competition sponsored by Satisfice

Mission

The primary mission of CAST 2006 is to help build an active community of software testing scholars, practitioners, and learners, and influence the software-testing practice through discussions among members of this community.

Registration

Registration is now open for CAST 2006. Prices are based on membership.

	Member Rates		Non-Members Rates		
	Regular *30% discount*	**Student** *50% discount*	**Early-Bird** *20% discount until April 30th*	**Regular**	**Student** *30% discount*
Conference + 1 day Tutorial *(June 5-7, 2006)*	$630.00	$450.00	$720.00	$900.00	$625.00
Conference Only *(June 6-7, 2006)*	$280.00	$200.00	$320.00	$400.00	$275.00
1-Day Tutorial Only *(June 5, 2006)*	$350.00	$250.00	$395.00	$500.00	$345.00

Register online at: http://www.associationforsoftwaretesting.org/conference/registration.html

About the AST

The Association for Software Testing is a nonprofit professional service organization dedicated to advancing the understanding and practice of software testing. The AST serves a community of scholars, students, and software development practitioners by providing forums for discussion of all aspects of software testing through conferences, publications, web sites, and other services.

Objectives

- Encourage, facilitate, and coordinate partnerships between testing practitioners and testing researchers.
- Publish the web-based AST Journal containing leading-edge articles on testing practice and theory.
- Host an annual AST Conference to bring together developers, testers, and researchers in an exchange of testing practices, theories, and techniques.
- Support the teaching of software testing by encouraging projects to develop and publish resources that assist classroom presentation, grading, and self-study.

Hotel Info

The Crowne Plaza at historic Union Station is housed within "America's first union station" and listed in the National Register of Historic Places. This Romanesque Revival-style hotel features 273 truly unique and spacious guest rooms, Pullman's Restaurant and Lounge and superb conference facilities.

Located in the heart of downtown Indianapolis, the Crowne Plaza is adjacent to the Indiana Convention Center and RCA Dome (home of the Indianapolis Colts). The Crowne Plaza is just moments away from everything including Conseco Fieldhouse (home of the Indianapolis Pacers), Circle Centre Mall, The Indianapolis Zoo & White River Gardens, Victory Field (home of the Indianapolis Indians), the NCAA Hall of Champions, and its minutes from a vast array of eateries, nightclubs, art venues & other attractions.

The Crowne Plaza Hotel is at:
123 W. Louisiana St., Indianapolis, IN 46225

Hotel Reservations: 1-877-227-6963)
See http://www.ichotelsgroup.com/h/d/cp/1/en/hd/inddt, for more hotel info & directions.

The Crowne Plaza Hotel CAST conference rate is $129.00 per night. Call 1-877-227-6963 to make your booking. Use the group code "SFT" get the conference rate via the Web-site.

Indianapolis International Airport

The airport is located 7 miles (11 km) from Indianapolis at:

2500 S. High School Road, Suite 10
Indianapolis, IN 46241
(317) 487-9594
http://www.indianapolisairport.com/

Transportation from the Airport

Taxi from the airport to the hotel (about $25 one way) or a shuttle ride (about $11 per person one way).

Pictures of Indianapolis are courtesy of the Indianapolis Convention & Visitors Association. Please visit: www.indy.org

Conference Organizers

Conference Chair
Michael Kelly

Program Co-Chairs
Jonathan Bach, Quardev Laboratories
Chang Liu, Ohio University

Program Committee
James Bach, Satisfice, Inc.
Scott Barber, PerfTestPlus, Inc.
Ross Collard, Collard & Company
Yuetang Deng, Google
Elfriede Dustin, Symantec Corp.
Danny R. Faught, Tejas Software Consulting
Arthur Goldberg, New York University
Matthew Heusser, Priority Health
Bernard Homes, T.E.S.S.C.O. Ltd, London, U.K.
Allen M. Johnson, Jr., Huston-Tillotson University
Cem Kaner, Florida Institute of Technology
Donglin Liang, University of Minnesota
Panagiotis K. Linos, Butler University
Johannes Mayer , University of Ulm, Germany
Grigori Melnik, University of Calgary, Canada
Henry Muccini, University of L'Aquila, Italy
Randy Rice, Rice Consulting Services
Brian Robinson, ABB Inc.
Vandana Shah, Agitar Software
Andy Tinkham, Florida Institute of Technology

Tutorial Co-Chairs
Karen Johnson
Ross Collard, Collard & Company

Publications Chair
Joanne Webb, Liquid Digital Media

Treasurer
Michael Goempel, Fusion Alliance

Marketing/Publicity Chair
Andrew Wetmore, X-Rite, Inc.

Local Arrangements Chair
Dana Spears, Number Six Software

Sponsorships Chair
Ross Collard, Collard & Company

Webmaster
Jonathan Kohl, Kohl Concepts Inc.