

Trip Report: STARWEST 2007

Introduction

I went down to California for Software Quality Engineering's STARWEST testing conference (<http://www.sqe.com/StarWest/Default.aspx>) which was held in Anaheim the week of October 22nd. Overall, it was a great week represented by people from a wide variety of industries. Below I outline some of the sessions and high level points that stood out in the talks which I attended.

Risk Based Security Testing

Presenter: Paco Hope (Cigital)

Before you start your security testing, you want to have an excellent requirements specification. Every detail in a feature of the product needs to be documented along with expected results for when the rules are broken.

Be sure to keep up to date on the latest exploits. By knowing what the current trends and attacks are, you will be better prepared to address and test for them in your product. Here are some sites and mailing lists to keep you informed.

- <http://www.securityfocus.com/archive/1>
- <https://lists.grok.org.uk/mailman/listinfo/full-disclosure>
- <http://www.securecoding.org/list/>

It is also important to frequent any sites that are dedicated to attacking your product.

When working on a threat model, be sure to start with the most important assets and data, then add in everything which accesses this information.

When dealing with (pseudo)random numbers, be sure to test or validate the randomness. Using scatter plots and/or statistics, you can easily spot non-random trends in the data.

Seven Pernicious Kingdoms

<http://www.cigital.com/papers/download/bsi11-taxonomy.pdf>

Interesting Tools

<http://curl.haxx.se/>

Suggested Readings

Anley, Chris et.al. The Shellcoder's Handbook. New York: Wiley, 2007.

Hoglund, Greg and Gary Mcgraw. Exploiting Software: How to Break Code. Boston: Addison-Wesley, 2004.

Howard, Michael and David Leblanc. Writing Secure Code. Redmond: Microsoft Press, 2003

Howard, Michael et.al. 19 Deadly Sins of Software Security. Emeryville, California: McGraw-Hill Osborne Media, 2005.

Whittaker, James and Herbert Thompson. How to Break Software Security. Boston: Pearson/Addison Wesley, 2004.

Root Cause Analysis

Presenter: Edward Weller (Integrated Productivity Solutions, LLC)

When analyzing a defect, we often investigate the issue, fix it, and then never think about it again. Ideally you take this to the next level and identify factors that can be put into practice which will prevent the issue from appearing again – on both the development side as well as on the test side.

Be sure to involve the original developer, tester, and program manager who initially worked on the code where the defect was found. They have the full background and understanding of the software system; which gives them the expertise needed to help define forward thinking solutions. Without knowing the reasons for why the project is the way it is, a group will most likely be unable to make the correct decisions.

Branching out using Classification Trees

Presenter: Julie Gardiner (Grove Consultants)

Julie talked about using trees to do test case pruning/prioritization, as a method for reducing a complex matrix of test cases into the minimal set of tests which hit all the important paths. She also demonstrated (using the below tool) the inverse, which generates the complete (exhaustive) list of all possible combinations.

Interesting Tools

http://www.systematic-testing.com/functional_testing/cte_main.php?cte=1

Ten Indispensable Tips for Performance Testing

Presenter: Gary Coil (IBM)

The presentation was focused on services, but a lot of the points are applicable to traditional products as well. Here are some of the lessons that deserve highlighting:

- Ensure you are testing in an environment that mimics what the customer will be using. Your results will be useless if you are running in a scaled down or simulated environment.
- Ensure the data the application or service utilizes is representative of the real world.
- Be sure to do both scalability tests (incrementally add load until it breaks) as well as endurance tests (run for an extended period of time with a sustained load looking for leaks, race conditions, etc.).
- Document everything and archive the results.

Selecting Mischief Makers: Vital Interviewing Skills

Presenter: Andy Bozman (Orthodyne Electronics)

This was probably the most interesting presentation I attended this week. I've thought a lot about how to best interview candidates, and Andy made some excellent points about the type of people you want to hire as testers: Hire mischief makers, not troublemakers.

Mischief makers typically work within the rules, but enjoy creatively pushing those rules to the limits.

During the interview you must first put the candidate at ease, then you must direct and control the conversation in order to pull out the type of information you are looking for. The clues will be very subtle, so you need to pay special attention to their body language, what words they use, and even what words they don't use when describing past experiences.

Here are some of the questions Andy likes to ask:

- Describe a memorable bug. Did the candidate talk about how he/she just followed a straight line case to discover the issue, or were they crafty in their approach.

- Describe an admirable trait from a previous co-worker. Is the candidates' model very straight-laced or was he/she a jokester?
- Describe a childhood prank. Was the event creative yet harmless, or was it boring or vindictive?

On the other hand, if someone is *too* mischievous, it can lead to trouble. These people are often bitter or resentful and will cause conflict with others or be disruptive to the project.

50 Ways to Improve Test Automation

Presenter: Mark Fewster (Grove Consultants)

Mark went into a plethora of lessons and best practices around test automation. He grouped the tips into the following categories:

- Planning and Management
- Scripting
- Comparison
- Pre and Post Processing
- Testware Architecture
- Testware Maintenance

All 50 lessons can be found here:

<http://www.grove.co.uk/downloads/generalPdfs/improvingAutomation.pdf>

Testing on the Toilet

Presenter: Bharat Mediratta and Antoine Picard (Google)

8:30 on Friday morning is probably not the best time to make the most of people's attention and awareness level, but Bharat and Antoine did an excellent job of keeping everyone awake. Their talk focused on Google's practice of placing technical articles in the office restrooms. Instead of staring at a blank wall while taking care of business, you might as well learn something new. A virtual team at Google took the grass-roots opportunity to write up a series of one-page whitepapers that described some technical aspect about testing, and then distributed them throughout the campus. The papers gave just enough information to be informative but not overwhelming, for topics that required more detail, a very simple URL was referenced on the page which could be looked up when you were back in the office.

Some of their (non-internal) papers can be found on their testing blog at:

<http://googletesting.blogspot.com/>

Expo

A good chunk of my time was spent at the Microsoft booth, where we were giving demos of Visual Studio, and showing off the Tester Center web site (<http://msdn.com/testercenter/>) which launched during StarWest. Overall people were pretty excited about the site, as there was definitely a feeling that bringing together testing experiences and knowledge from across the industry is an excellent way to enhance the overall quality of software and improve testers' methods and technical ability. For questions about submitting your own content to the Tester Center, send email to tcsuubmit@microsoft.com.