Company: Allion Test Labs
Intern: Russell Miller
Mentor: Chad Meyer
Mentor Phone: 509-995-4444
Mentor E-mail: chadmeyer@us.allion.com
Discipline: Computer Science
Internship Level: Junior Year
August 30, 2011

About Allion

Allion was established in 1991. Through custom test plans and automation, Allion offers engineering services for performance and compliance testing. The specialty of the lab in Beaverton is advanced Wi-Fi testing. For performance testing, there is an anechoic chamber and other equipment that allows isolation of signal. Using attenuation simulators, several use cases can be tested in-depth.

My Role at Allion

Projects at Allion coincide with contracts from third parties. Often these projects involve testing a new product from those companies. I worked on a couple of those projects, but worked on side projects as well when we were between contracts. I developed automation for the attenuation equipment, and wrote full tests that used that automation. My mentor Chad was extremely helpful, always willing to teach me about wireless technology and testing.

Rather than having departments Allion is split into teams per project. When I first started my internship I started out helping analyze results from the last round of testing for the Nook Color, when it was still pre-release. When that project ended I did independent work until another testing contract started later on in my internship.

The Projects I Participated In

- New product testing for Barnes & Noble's Nook Color
- Writing a Python library for the Azimuth Adept-N
- Writing a Python library for the Azimuth Ace
- Writing Automated Testing which employs the Adept-N and Ace
- Super Top Secret new product testing from a new startup company
- Super Top Secret new product testing for Amazon

Executive Summary

On the Nook Color, I was doing an investigation of the data we gathered, attempting to provide the value-add of knowing why certain tests failed. The utility and testing libraries I wrote involved doing research on the equipment's API, gathering requirements from my mentor and engineers on site, maintaining these libraries over the course of the internship, and writing documentation for engineers to allow them to run my scripts with ease. With the secret projects where we tested new products, I was able to write the test methods to meet the needs of the test plan we had provided to the company that was developing the new product. I wrote several scripts and ran them over a large number of iterations, as the test plan called for. There was a team of engineerings using the scripts I wrote, and I had to describe the script to my team.

The chief export of Allion is the ability to produce the tests outlined in a test plan, which is agreed upon by the company paying for the contract. On a regular basis, we keep the customer updated on completion of the test plan, and we were able to impress our customers with timely results and added value of debugging problematic scenarios. On one of our secret projects we were given the opportunity to write bug reports to their database, and I was able to benefit not only the customer but the face of Allion by paying attention to detail and displaying intuitive knowledge of the functionality of their device.

The greatest personal achievement of mine was deploying the first release of the automation scripts I wrote. They controlled the attenuation equipment one of our engineers was trained to operate. Previously he spent much of his time running tests that had a lot of interaction with a computer application that changed signal

attenuation throughout a test. Because of my work, he was able to simply configure a test, start it, and look at the results when it completed three or more hours later. He was so grateful, and expressed so much gratititude, that I felt like I had really made a difference for the first time in my Computer Science career. This made it a joy to provide him with new features and maintain these libraries and scripts later on. He was always very understanding when bugs were discovered, and offered great feedback when I had written patches.

More About the Azimuth Libraries

As mentioned above, I was able to provide automation to one of our engineers that cut down on very tedious work. The development process, however, was an interesting one. When the idea was first presented to me I was simply told "We think this can be done in Python. Figure it out."

A PDF came with the equipment that had a "Programmer's Guide" with a few commands that a *programmer* can issue. I knew that all I needed to do was set the attenuation, and possibly check what it's currently set at. I found the commands that did this, but I had no idea how to issue them.

I discovered that this API was designed for TCL, which is just another scripting language. I had never seen or used TCL before, but I kept at it and got these commands to work. I ended up writing simple scripts that could be called from the command line. This allowed us to connect to the control PC that's attached to the Azimuth equipment and run the necessary commands. I wrote the Python libraries around this concept.

The way the network infrastructure is set up at Allion, combined with the libraries already being used in testing, it worked well to import the library I had written for attenuation control and write tests that controlled attenuation via my simple API. I wrote some simple scripts that demonstrated the fact that this actually worked in a reliable way and showed it to my mentor and some engineers around the shop. I talked to them more about how this could be used in future testing, and that's how the testing libraries were introduced to me.

From there I learned how we loaded tests into a queue, and how I could write a compatible test script that could be loaded and automate the attenuation controls. A lot of work was involved, because things always seem to change once all of the pieces are working together in a test. But after a lot of communication and (sometimes frustrating) debugging, I had reliable test scripts written that are still being used to this day.

More About Testing a New Product

Due to the top secret nature of the project I was the test developer for, I have to explain this with care. We wrote a test plan before we ever saw this device or even knew what it was supposed to do. In fact, throughout the project we barely ever discovered what its use cases would be. However, the device ran an SSH server and had a serial connection, needed wireless testing, and was expected to be compliant on a range of wireless access point manufacturers.

To start I wrote up some mock scripts that simulated how the tests would run were the device simply a computer running Linux. These tests included putting the device into a suspend state, waking the device from a suspend state, and using pings to verify that the device is in the state it should be. Once I had worked out the kinks in developing these simple scripts with a Linux computer, the only thing that needed to change was how the states changes were implemented. Using SSH to connect would be the same, as would checking the state with pings.

Because these scripts were ready to go before we ever saw the product, we were able to present results within the first day of the customer coming to our shop to present it to us. This got us off on the right foot in their eye, and opened a channel of communication between me and the company representative - since I was actively working with their device and finding out how things worked in order to run these tests.

One of the biggest challenges of this project was constant change of requirements. They appreciated my hard work and my detailed reports, but they continually changed priority of what was being tested. For example,

our test plans seem to always focus on performance and we never covered that with this project. We spent the majority of the effort on interoperability - which is basically testing compatibility with a very wide range of AP/Router models.

The most significant example of a requirements change that took place several times was with a test where we were suspending the device and attempting to wake it up after a certain amount of time had passed. This can be tricky because the device isn't doing any communication while in a suspended state. Not only that, but the company had developed their own implementation of the wakeup signal. The scripts I wrote for this project went into a version-controlled repository, and the number of revisions that have been committed is well over 200 now. Each time the device was updated, or test expectations changed, communication was key in getting future tests to run smoothly. Hundreds of emails were sent, there was a lot of interaction via their bug database, and multiple conference calls took place. I worked with a fantastic team of engineers that were running my scripts and reporting bugs, and many of the issues we ran into were solved only because we did it as a team.

In Conclusion

The thing I learned most about during this internship is communication. Within a team, or with a supervisor, or with a customer, it is extremely important to communicate well. On the team level, it is extremely important that everyone is on the same page. We had to work in stride without stepping on each other's toes. We used things like collaboration software to stay in sync with technical info such as who is running what test, how our equipment is configured, which wireless channel we're using, and what tests still need to be done. We help each other stay up to date on information about software updates and changes to the test methods. Sometimes setting up for a test or running a test doesn't quite work, and we help each other fix it.

Communication with my mentor has been extremely rewarding. When I got here I knew nothing about wireless technology or how it is tested. I have learned so much from his demonstrations, and he does a great job of illustrating his points on the dry erase windows we have in our labs. Once I feel like I've met the requirements of a test method, I run it by him again to double check and we talk about the results afterward. Many times doing so has allowed us to improve the test method so that we can provide even better results.

When given the chance to talk directly to a customer about the tests or their product, I've practiced a very diplomatic reserve. There have been features and bugs that have frustrated me greatly and I have to simply observe and report. What I really want to do is tell them they're doing it wrong, but that would obviously not be constructive.

Allion's scripts and libraries are written in Python. I had a growing passion for working with Python before I started this internship, but I did not have a very broad knowledge of the language. Throughout this internship I have been constantly learning new ways of implementing tests and writing Python packages. Russell, the senior software developer at the shop, has given me some excellent instruction. His passive yet encouraging feedback is always helpful. He doesn't ever come out and tell me I'm doing something wrong, but he's introduced a lot of new concepts to me. Mimicking his style has helped me grow, and I feel confident about the deployment and documentation of my work.

The primary goal I set for myself was to learn about software development on a larger scale, and to learn how a software developer interacts with fellow developers and customers. The scale of our projects wasn't very large, but I have at least worked on some development within a team. The goal that I feel I am still pursuing is to do development of a software product for a customer. The development I did here was scripts that the engineers and I were running. The meta product that was delivered to the customer was a byproduct of those scripts. While I appreciate everything I've learned here, the experiences I've had, and the people I've gotten to work with, I still feel like there is much more for me to learn.

As mentioned before, there are two main benefits that I think Allion gained from my work. The libraries I wrote and automation I developed will help with many, many projects down the line. I really hope that when I'm not around the code can be maintained without too much trouble. I've attempted to write as much documentation as I could to help with that. The other is with the testing we've been able to do for these

projects. I've shown that I can get things working in a pinch and really impress the customer. More importantly, perhaps, is the effort I've given to communicating results to the customer. With the project I spent the most work on, I really developed a relationship with the company that allows good communication between our engineers and theirs. Our contract has been extended multiple times, and while that definitely has to do with our management's abilities and a lot of really good engineers, I do feel that I played a part in that as well. I really began to feel this way when the representative from their company specifically requested that I continue to work on the project past one of the extension points.

Buzz Words

The following is in addition to terminology associated with Linux and networking.

- DUT - Device Under Test
- AP - Access Point
- RVR - Rate vs. Range
- OTA - Over the Air
- Cops - Continuous Operations
- ND&S - Network Detection & Selection
- Cell Center/Middle/Edge