TauNet Project Evaluation Report

Copyright © 2015 Matthew Tighe

Table of Contents

1. My process

I started TauNet by creating the system requirements specifications. Later, once I had a slightly better idea of the system I was going to create, I edited it to better reflect what I thought would be the final product. I also wrote the design document with this same idea in mind. Before I started the implementation, I wrote a brief checklist of things that I thought would be important to test during implementation and after the code was finished. Then I implemented CipherSaber 2, the server, and the client, roughly in that order. Once those were finished, I updated my documents to better reflect the final product.

2. Results of my process

The creation of the SRS and SDD were both massively helpful in understanding what exactly I was creating. Although the system I originally envisioned didn't end up being the complete implementation, having guidelines eased the process of coding hugely. Eventually, my process resulted in a working (though slightly bare) implementation of the TauNet system.

3. Lessons learned

Lessons on SRS creation:

Spend more time here. Having a more complete understanding of the project would have been massively useful during implementation. The more time spent identifying requirements, the better.

Lessons on SDD creation:

Same idea as SRS. Had I better designed what I planned to implement from the get-go, I can only imagine that implementation would have gone much more smoothly. Much of the interface documentation I put in the SRS would probably be better suited to the SDD.

Lessons on STP creation:

Figure out input/output pairs early on. Decide what can be done during a visual inspection, and what needs to be more rigorously tested with methods like automated tests. In the future, I'd also like to be more aware of different testing methods and how they could be useful on those future projects. Be more aware of how to break implementation into good, clean units.

Lessons on implementation:

Write better pseudocode. Understand algorithms central to system performance (like Ciphersaber in this case) more completely before you start trying to implement them. Have your units worked out before you start trying to implement a project.

Ultimately, spend more time on actually engineering software so that implementing it becomes easier.

4. **Future desires**

If I end up continuing my work on TauNet, here are some things I'd like to include:

- A secure way to store messages
- A way to check keys without hardcoding them into the implementation
- Change the server to only receive messages and store them
- Change the client to be able to display a list of messages and display those messages
- Download the user list automatically