# **Senior Project Final Self-Assessment**

This document is intended as a guide for the senior project team to assess its performance in a number of dimensions. You need not answer each question in detail, rather, use the questions as a guide for the kinds of items to assess. Add items you feel are appropriate.

This self-assessment will be one of multiple elements that your faculty coach uses to arrive at an assessment of the team’s performance for this second term. The other elements that the faculty coach will use include: direct observation of the team, team peer evaluations, reviews by other faculty during the project presentation, sponsor evaluation, and project deliverables. These self-assessments will also be used as part of the SE program’s accreditation and curriculum improvement efforts.

To complete this self-assessment the team should carefully consider each of the questions and provide an honest evaluation of the team’s performance. Your faculty coach will inform you when this self-assessment is due and how to deliver it.

### Team: Cobra

### Project: Ouroboros

### Sponsor: Nathan Ransom

### Faculty Coach: Rick Weil

### Product

1. Did the team prepare all the documentation artifacts requested by your faculty coach and sponsor? Were these documents carefully inspected prior to delivery? How would you assess the quality of the document artifacts?

The team has completed all required documentation and artifacts as required by the sponsor and faculty coach. These documents were were inspected before their submissions, and the team believes them to be of high quality.

2. How well did the team elicit the requirements? What approaches were used to elicit the requirements? Were key requirements missed? What methodology was used to document and validate the project requirements?

The team did a fairly good and thorough job eliciting requirements at the beginning of the project. Requirements were gathered through sponsor discussions about the scope and behavior of the system. As for documentation and validation, requirements were recorded in an SRS document. This was then sent to the sponsor for approval before major project work began. For validation, the team broke up the requirements into major features that needed to be completed, and kept track of their status and progression as the project progressed, ensuring that these features that mapped to requirements were completed.

3. Did the team explore the entire design space before arriving at a final design? Have there been many errors found in the design? Was it necessary to make major changes to any part of the design? What were the reasons for the change?

The project design was mostly presented to us by the sponsor. While it is true that some areas have changed over the course of the project, for the most part the design has remained the same.

4. How has the development and implementation progressed? What percentage of the product do you estimate was completed? Is the team providing the documentation within the implementation artifacts?

Development has progressed smoothly and continuously over the course of the project. At this point we have met 100% of the customer’s requirements and some stretch goals as well. The team has also included documentation on the design, usage and installation process for the end deliverable.

5. What was the team’s testing strategy? Did the team develop a test plan? If so, was it followed? Did the team performing unit testing? Did the team use any test frameworks, such as JUnit? What are the testing results? Were any major defects found during system test? If so, were they fixed? Did the team do regression testing?

The testing strategy for the project was conducted on two fronts. Unit tests were constructed for the backend code and checked before new changes were merged. For the rest of the system, pre-defined XML documents were passed through the generator. The system output was compared against predicted output to ensure project integrity.

6. Products need to be designed within guidelines and constraints appropriate for each project. It is also important to consider the impacts of the products that are designed. In the following categories discuss the constraints and impacts that have a bearing on your project. Note that all of these categories may not have bearing on your project but your project is probably affected by many of them.

· Economic issues

Aside from being sold in a future revision our project has no impact on the economy.

· Environmental issues

Our project has no environmental impact.

· Social issues

Our project has no social impact.

· Political issues

Our project has no political impact.

· Ethical issues

Our project has no ethical impact.

· Health and safety

Our project has no health to safety impact.

· Manufacturability

Our project has no manufacturing impact.

· Sustainability

Since our project is open source, it will be sustained so long as the community chooses to keep it updated.

7. What industry and engineering standards was your project required to adhere to? Were these new standards that the team had to learn? Did your sponsor provide you support for understanding these standards? Did you have to educate your sponsor about these standards?

For the embedded server we were required to use Mongoose with C++ 03/98. Being used to newer standards, we had to find out how to implement missing functionality or work with external libraries. The sponsor answered all questions that were asked, but most of our work included learning these new libraries on our own.

### Process

1. What was your process methodology? Was the process appropriate for the project? Did you follow the process or modify it as the project progressed? If you could repeat the project, what would you do differently?

The team implemented an iterative methodology, which was appropriate for our project since our customer wanted to see working parts of the system as they were developed. By iterating through the construction of the system, the team was able to show progress to the customer. The process changed a little as the team transitioned from one semester to the next. The general idea of having one major iteration per semester remained the same (having a working system by the end of each major iteration), but the team began implementing weekly iterations in the second semester, where each iteration would consist of the team reviewing plans for the week, then speaking about any blockers, and then implementing the work. In short, if we were to start over again, we would do everything we did this semester from the start. Weekly updates, less focus on time tracking, and not using unnecessary issue tracking software such as JIRA should have never been used.

2. Was there a large requirement to learn the problem domain? What approach was used to gain domain expertise? Did your sponsor provide adequately support? What forms of support did you receive?

There was a large upfront requirement to learn the problem domain. A lot of meetings at the beginning of the semester were spent talking about the project architecture. A lot of those meetings were spent with the sponsor eliciting requirements once we clarified our misunderstandings of what the system should do.

3. What mechanisms did the team using to track project progress? Did they give the team and sponsor adequate insight into project progress and issues? How well did the team track its project progress? How often did these artifacts get updated on the department project website?

The team made full use of GitHub for tracking project process. Using the issues tracker, everything was associated with a milestone to track how far along the team was with each part of the project. This repository was public and shared with both our coach and the sponsor so that they could check on our progress at any time. Every time a Pull Request was reviewed and accepted they were tagged so that the issues would be automatically closed as well.

4. Did the team conduct effective meetings?

Meetings were effective when needed, but meeting twice a week made it hard to always have effective meetings. During busy times (normally at the end of the year) we chose to meet on the weekends as well to ensure all work was done.

5. Did the team meet all project milestones? Which milestones, if any, were missed or were met ahead of schedule? What contributed to schedule changes? What could the team have done differently to ensure that milestones were met?

The team met most project milestones. Originally the team wished to have more features in for the end of the first iteration, but these were completed early in the second semester. The delay was attributed to over-ambition, but to clarify these expectations were above that of the sponsor and the sponsor was pleased with our results.

6. Was the team required to adopt new technologies? What were these technologies? What approach did the team use for selecting the appropriate technology for the project? Did the sponsor provide any support for learning these technologies? How well did the team ramp up on the new technologies and begin to apply them effectively?

The team was required to use the Mongoose web server as the backend for the system. Team was also required to allow for users of the system to provide some form of configuration and then have the system generate code, for which the team was suggested to use XML for the configuration language and Ruby rgen for the code generation aspect. The final executable was generated via compilation through Autotools. The working system system also had to expose both a REST and a C++ API. The team decided to use the suggested technology choices. In terms of learning the technologies, the team split up the components of the project amongst the team members, who then specialized in the technologies used in their components and experimented with them early in the first semester in order to familiarize themselves with the technologies. The sponsor provided insight with regards to his previous experience with the technologies the team used and warned the team of possible issues and traps. By the beginning of the second semester, the team understood the technologies well enough to use them without them getting in the way of development.

Another aspect of the system the team had to learn, related to the REST API, was working with JSON. The team was able to leverage the information provided via the REST API via JSON objects to dynamically generate the web UI, with the help of JavaScript and Bootstrap.

7. How well did the team maintain quality control over the project artifacts? Have all artifacts been reviewed for adherence to quality standards? What was the review process used by the team?

Everything was made into a Pull Request where it was code reviewed by several team members. Additionally, at least one teammate pulled down the code and verified that it worked after manually testing it. Only then was the request accepted.

Other artifacts were reviewed in meetings before being OK’d by the group for submission.

8. Did the team have any issues with configuration management? How were these problems solved? What percentage of project artifacts is under configuration control?

The team cannot feasibly address all possible target platforms Ouroboros can work on, and as such that kind of testing was out of scope. Autotools was used for helping with compilation, but it had its own problems, such as not being very friendly with different versions of itself. Developers using the project are encouraged to use the versions mentioned in the technical report.

9. What was the set of metrics that the team tracked? Did the team gather these metrics on a consistent basis? What did the team learn from the review of these metrics?

The team kept track of hours worked on different roles in the project as well as punchcards to determine when development took place (commit times). After the first semester, time tracking was unhelpful, and diligence in reporting time fell off. From commit times in the punchcards, after the first semester it was determined that little to no development was being done on the weekends. From this insight we changed our process to help encourage more consistent development on the weekend.

### Communication and Interaction

1. How well did the team communicate project progress to the sponsor? What regular communication did the team have with the sponsor? Did the team been maintain this communication to the satisfaction of the sponsor? Were any adjustments needed in the communication over time? Were these changes initiated by the team or the sponsor?

The team communicated by email once a week to the sponsor (there were some exceptions) with updates of what each team member was working along with the overall progress of the project. On a non-regular basis, the team met with the sponsor once every two weeks to demo anything that could be shown to collect feedback as well as to get guidance on how the sponsor would like certain functionality to work. For two stretches of time the communication was not to the sponsor’s preference, once in each semester and one of the times partly due to miscommunications on the sponsor’s travel plans. After the feedback of lack of insight in the first semester, the team began to send out the weekly email updates to make sure that there was more transparency in the progress of the team (this was the team’s idea which the sponsor was happy with).

Communication eventually broke down for many reasons (sponsor went away on a business trip, team became bogged down with other work, etc.).The takeaway regarding communication is that the team should have used the sponsor’s preferred method of communication more extensively (email, in this case) and use more detail in communication (e.g. put git digests/summaries along with emails).

2. Did the team need to provide technical input to the sponsor? How well did the team educate the customer in these areas? What mechanism did the team use?

The team did provide technical insight on the project to the sponsor. Our project sponsor was a software developer for Harris and an instructor at RIT so communicating with him on a technical level was not challenging. Email was used most often as a form of communication as per request of the sponsor, although the team tried to meet biweekly with the sponsor in the SE team rooms.

3. Was this an effective team? What has been contributing to and detracting from the team’s effectiveness? What are the team’s weak points? What are the team’s strong points? What changes could the team have made to make it more effective?

The team was effective. The effectiveness of the team was lessened by other classes. What made the team strong was when a team member was freed up, they would catch up. This trait in the team allowed for a dynamic workload that helped get the work done (usually) before deadlines were met.

Where the team could have done better, as described earlier was communication. Another point that could have been improved was the accuracy and frequency of metric tracking. For tracking our metrics, one possible improvement would have been to add some sort of hooks into the commit process that would force the team to update our metrics.

4. What mechanism did the team use to communicate with the faculty coach? Was communication with the coach effective? Were there any trouble spots with the faculty coach communications? What could the team or faculty coach have changed to make their communication more effective?

Email and in person meeting were most often used as a form of communication with the faculty coach. From our perspective there were no challenges or issues with communicating with the faculty coach.

5. Did the team need to interact with department staff personnel, i.e., the office staff or system administration? Was this been handled in a professional manner? Were there any problems with these interactions?

For the most part, the team did not have to interact with department staff personnel. From the Software Engineering department, the team’s only interaction was to request a server, and from the sponsor’s side, the team was invited once to go to the Harris building for a tour, followed by a quick discussion of the state of the project with the customer and another colleague of the customer. The team did not have problems with professionalism while interacting with the customer or other parties.

6. Does the team have a complete website with all project artifacts stored and up-to-date on the software engineering department webserver? How often were entries on the webserver updated?

Not currently but a final commit will be made once a few lingering pull requests are resolved in our final meeting. Entries were made to the web server exactly twice, once at the end of fall and once at the end of spring.

7. How well has the team made presentations to the sponsor and faculty coach? Was the final project presentation done in a professional manner? Was the poster presentation done in a professional manner? What could have been done to improve the team’s presentations?

The team believes that we have not had problems presenting to our sponsor and faculty coach. Some of the audience members in the presentations believed the team could have dressed up more and/or more consistently, which technically could be true, but our faculty coach was OK with our manner of dress for our presentations. The team received positive feedback about the poster. If there is one thing the team could have done to improve both the presentation and the poster, it would be to start making/planning them sooner than they actually were.

8. Does the technical report adequately document the project and its results? Was the paper of high technical and editorial (language, style, grammar, etc.) quality? Did all teammates contribute to the paper? Did the sponsor contribute to the paper? Did the sponsor review the paper?

The technical report goes into quite a bit of detail with regards to the inner workings of the different components of the system. The team believes it to be of relatively high technical and editorial quality. All team members contributed to the paper, and all of them contributed to its review. Due to how late in the process the team started the paper, the sponsor did not have a chance to contribute nor review the paper-- the team’s faculty coach did have a chance to review it.

9. How well did the team work with other senior project teams, coordinating access to lab space and equipment, sharing experiences and ideas, etc.?

The team rooms were on a first come, first serve basis. It would be beneficial to have enough team rooms for everyone. Our team coordinated with other teams to keep up to date on project deliverable deadlines and to troubleshoot problems with our own respective projects. Experiences within and between our sponsors were also shared with other teams.

### Achieving Customer Satisfaction

1. In the team’s opinion did the work satisfy the project sponsor? Were there any weak spots in this regard?

In the team’s opinion we fully satisfied the sponsor. This was based on the direct feedback of the sponsor. Weak spots may include communication during the work between the team and the sponsor. At some points the sponsor was not updated on the current status of our work and was unsure of our progress.

### Achieving Team Satisfaction

1. Did the project satisfy the team’s expectations for learning? Were there any weak spots in this regard? What could have been done differently to improve the team’s learning experience?

The project was very interesting to the team, it incorporated many components that played to the strengths of the team while allowing us to individually explore new areas we were unfamiliar to. To improve our learning experience, the team could have been given more freedom to explore other technologies for implementation of the project.