# **Senior Project Interim 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 first 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 interim project presentation, sponsor evaluation. These self-assessments will also be used as part of the SE program’s accreditation effort.

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: Harris

### 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?*

Yes, the team did prepare all documentation artifacts requested by the faculty coach and sponsor. For the most part, these documents were carefully inspected prior to delivery. As they currently stand, the documents are of moderate to good quality, but some effort should be spent making sure that these are still up to date, since these have not been updated in some time.

*2. How well did the team elicit the requirements? Are the requirements fully specified at this point? What approaches were used to elicit the requirements? Were key requirements missed? What methodology was used to document and validate the project requirements?*

Requirements are fully specified at this point. In order to elicit requirements, the group convened and did our best to understand the original proposal. From there, we developed a set of questions about the project regarding the properties of the system, and effectively interviewed the sponsor in order to gather the requirements of the project. Afterwards, we collected the information we had acquired and made the documentation for the project (Vision and Scope and SRS documents), which we presented to the sponsor to make sure that we understood the requirements. This document covered the majority of the sponsors requirements for the system. At this point some unknown aspects about the system remained but were limited to aesthetics and style concerns. The team used a rapid prototyping and feedback cycle with the sponsor to gather and gauge effectiveness on meeting these final requirements. To our knowledge, no key requirements were missed.

*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? Do you have a complete design at this point?*

We do not believe we explored the entire design space before arriving at our current design, but we did spend a significant amount of time in the documentation and design phase of our process.

We have a relatively complete design at this point, with the only aspects of the design fluctuating at the moment being implementation design choices, which change as our understanding of the limitations of the technologies we are using improves. The general design appears to be stable.

*4. How has the development and implementation progressed? What percentage of the product do you estimate is complete at this point? Is the team providing the documentation within the implementation artifacts?*

Implementation and development appears to be progressing at a brisk pace. Based on input from our sponsor and faculty advisor, the team spent too long in the documentation and design phases of our process, but it appears that we are currently on track to complete the project by the end of the second semester. The current implementation of the system is a thin, fully functional vertical slice of the system and covers 40% or so of the proposed final product. Implementation plans for next semester are to widen out existing sections of the codebase with relatively little new design work being done. With regards to documentation within the implementation artifacts, these will be developed and provided. Currently only some documentation is available, but we hope to be able to expand the documentation of the implementation to the point of documenting even the generated results of the system.

*5. What is the team’s testing strategy? Has the team developed a test plan? Is the team performing unit testing? Is the team using any test frameworks, such as JUnit? What are the testing results to date? Were any major defects found during system test?*

The team is utilizing unit testing for the C++ portion of the project, using googletest as the framework. Unit tests are far and few currently, and will be built up over break and in the first weeks of the next semester. Next semester we are planning to work in integration tests as well as some way to test the automated tests generated by the system. Unit testing on the C++ back end has helped to identify errors, including one with one of the libraries we are using for handling regular expressions in C++. As for other testing, there has been some mention of implementing integration testing, but more discussion needs to be had on this matter in order to set it up.

*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 there may be one or two categories that have no bearing on your project but your project is probably affected by almost all of these.*

**Economic issues**

Our project is being developed and released under the MIT open source license (with the exception of the Mongoose web server we are using, which is released under the GPL v2), so it is freely accessible for anyone that can download it. It is expected that this project will save Harris development time, and thus money, when prototyping their radios.

***Environmental issues***

Other than perhaps boosting development prototyping speeds for servers in embedded devices, we do not see any environmental impacts due to our project.

**Social issues**

We do not see any social issues or impacts arising from our project.

**Political issues**

We do not see any political issues or impacts arising from our project.

**Ethical issues**

The only point of interest here is that our project uses a mixture of open source licenses. The project itself is licensed with the MIT license, but a sub portion of it requires the mongoose server, which is licensed under the GPL v2. Users of our projects linking to Mongoose need to be aware of their legal responsibilities should they choose to distribute their binaries.

**Health and safety**

It is possible that someone may want to use our project and deploy a server made from it and use it in embedded health and safety infrastructure. This would not be a good outcome, since our code is not up to the standard necessary for it to be used in health and safety situations. Moving forward, we may want to place a notice of warning in our project regarding this point.

**Manufacturability**

Our project itself does not require any special hardware, beyond that of a basic computer that can run an operating system that can run the mongoose server and that has Ruby available. On the other hand, it is very possible that our project can be used to generate and install simple web servers in embedded devices to be used commercially.

**Sustainability**

The team is aware that they will not be the last group tasked with writing code for this system. Keeping that in mind, the given project has been designed so that future developers will be able to add or modify functionality with little to no impact on the rest of the system by using established design patterns and documenting the project. Due to this, we do not see any sustainability issues arising from our project.

*7. What industry and engineering standards must your project 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?*

Our project did not have any industry engineering standards it was required to adhere to.

### Process

*1. What is your process methodology? Has this been clearly outlined to your sponsor and received the sponsor’s approval? How is the process documented?*

Our process methodology is the iterative process. We felt that this process would allow us to create a thin vertical slice of the system fairly quickly and would give our sponsor an opportunity to see progress and give us input as to how we should proceed. We have spoken with our sponsor and received his approval. We have set up the iterations as one major iteration of the development lifecycle per semester. We are keeping track of our process by documenting our time spent in the different activities on a time sheet, making tasks and issues on GitHub, and using our meeting time during the semester to coordinate our efforts. Moving forward, it may be beneficial for the team to define more formally how we plan on using GitHub for keeping track of our process, since this is something we began leveraging more heavily towards the very end of the first iteration.

*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 relatively large amount of technologies that the group had to become acquainted with. Part of the problem is that the group came from different backgrounds (two were more focused on web development, two come from more of a hardware background). The project work was split up among the team members according to what they were most experienced with, and then in meetings we reviewed what we had learned, spreading the knowledge around. The sponsor suggested technologies for us to look into, and we, as a group, did so, and from our experimentation and prototyping decided which ones to use and which ones to discard.

The main source of domain expertise was developed via the construction of prototypes and verifying with the sponsor that the prototypes were headed in the desired direction.

*3. What mechanisms is the team using to track project progress? How well has the team tracked its project progress? How often do these artifacts get updated on the department project website?*

The main mechanism the team used to track project progress was keeping track of time spent and on what in a time sheet one of the team members designed. Also, as of the end of the semester, the team has begun to use GitHub more actively in order to track pending tasks, issues, questions, etc. Overall, the team’s efforts to track progress have fluctuated over the semester, but on average we have updated our time sheet consistently. As mentioned earlier, task and issue tracking was weak at the beginning, but is now beginning to be more clear by an effort from the group to leverage GitHub’s functionality to do this.

In terms of the artifacts on the department project website, we have updated all of our documents and uploaded them to the documents section of our website.

*4. Is the team conducting effective meetings? What can be changed to make the team meetings more productive?*

Our team meetings have been fairly effective for the most part. We usually start each meeting discussing logistics and upcoming deliverables. After that, we discuss what each of us has worked on and show each other the progress we have made in development. This has been going well but to keep everyone more in tune with how the system is coming together, more in depth code reviews might become useful to us especially as the code base becomes more complex.

*5. Has the team met all project milestones to date? Which milestones, if any, were missed or were met ahead of schedule? What contributed to this schedule changes? What will the team do differently to ensure that future milestones are met?*

Our first major milestone was to have an end to end working prototype. This was met early on in the development process in an effective manner, allowing us to receive useful feedback from our sponsor. Our second major milestone, set for the end of the semester, was to refactor the prototype for our first official release. This milestone was not met to our complete satisfaction on time but is not too far off from being completed. We expect to have it completed some time before the beginning of the second semester of classes.

For future milestones, the weekend standup meetings we are implementing should help the team remember to work on the project more than two days a week, which should help increase the amount of work being done, as well as further improve group communication. Moving onto the second semester, we are planning on tracking our work better on GitHub, which should facilitate collaboration and problem solving to make sure development is never halted.

*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?*

Yes, this project has required us to learn the basics of code generation. While this is not a technology in as of itself, it required an adjustment to the way we thought about developing. We also had to choose a language for the actual code generation and our sponsor helped us decide on Ruby. After some more research into code generation, we were able to decide on a gem called RGEN to assist us. In terms of configuration files used as input for the code generation, we were asked to use XML, which required for us to learn how to write XML and also how to develop XML Schemas for validation. Experimentation and prototyping helped us quickly gain knowledge into how to use RGEN and work with XML.

We also had to learn how to build an embedded server in C++. This was something fairly new to all of us, although some of us were more experienced in C++ than others so the others had a bit more catching up to do.

The sponsor was always willing to help us with any confusion we might have had when getting started and deciding how to move forward with development.

*7. How well has the team maintained quality control over the project artifacts? Have all artifacts been reviewed for adherence to quality standards? What is the review process used by the team?*

The team has employed code reviews in order to make sure that the code merged into the master branch of our repository is of a good enough quality. With regards to the other artifacts, such as the SRS and Vision and Scope document, all of the team was asked to review each document, make comments, and then met to discuss the comments and make a final review before sending the documents over to the sponsor for approval, from who sometimes we received extra commentary which we analyzed and incorporated into the document, if necessary. To our knowledge all artifacts have been reviewed at least once, but it would be beneficial to review them again soon in order to make sure all artifacts are up to date.

In order to review submitted code we use GitHub’s pull request feature, where new features are developed in separate branches and then merged into the master branch after a code review. Our current code review process could use some fine-tuning, but right now all developers read through the changes, make notes and observations and questions, and once all concerns are addressed, and once an agreement has been reached in the team, the pull request is closed and the feature branch is merged into the master branch.

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

For this project the limit of configuration management we need to do is to keep track of the dependencies of the technologies we are using for generating code and for compiling the end product. Since this project is expected to be deployed to many different platforms, the sponsor told us not to worry about the specifics, so long as we kept track of the dependencies of the different components of the project. It would be up to the user of the project to fulfill the dependencies.

*9. What is the set of metrics that the team is tracking? Has the team gathered these metrics on a consistent basis? What has the team learned from the review of these metrics?*

From tracking the kind of work we did in the hourly activity tracker, we learned that ~29% of our time was spent in development. We thought this metric was slightly unrepresentative because some of our time spent researching different technologies was marked as development when it was to help make more educated decisions during our design process.

Another metric we watched was the git punchcard for commits. From this metric we found that we didn’t have as many contributions on the weekends, and they tended to be bunched more between tuesday and thursday (during and in-between our meetings). To solve this we will have checkins in a dedicated slack channel on the weekends to keep it as top-of-mind while we do our other schoolwork.

### Communication and Interaction

*1. How well has the team been communicating project progress to the sponsor? What regular communication does the team have with the sponsor? Has the team been maintaining 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?*

Our communication with the sponsor should’ve been more frequent and slightly more organized. We met with him every two weeks, but towards the holiday’s our meetings were less frequent. Also, while we did provide agendas for most of our meetings, we didn’t send these out until just a few hours before the meetings, something our sponsor recently mentioned seemed a little late. End of semester post-mortems revealed the sponsor would like more communication during the second half of the project. Plans for next semester involve weekly status emails as well as more frequent communication about project status.

*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?*

Our sponsor is a technical person himself so educating him on technologies for the project was not an issue. He actually helped point us in the right direction during the initial requirements gathering phase in regards to the technology we should use.

*3. Is 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 can the team make for next term that will make it more effective?*

Team Cobra is an effective team, as indicated by the work it has done on Ouroboros. An area for improvement for us would be to keep our sponsor updated more frequently on our progress so that we can receive the necessary feedback we need in order to move forward in the best direction. We also can improve our workflow for merging our code. Our strong points are mainly in regards to how adaptive we’ve been to learning new technologies and our initiative to produce results.

*4. What mechanism does the team use to communicate with the faculty coach? Has communication with the coach been effective? Are there any trouble spots with the faculty coach communications? What can the team change for next term to make their communication to the faculty coach more effective? What can the faculty coach change to make his or her interaction with the team more effective?*

First question: The team uses email to communicate with the faculty coach.

Second question: Communication between Team Cobra and it’s faculty coach has been effective using email.

Third question: There are no ‘trouble spots’ with the faculty coach.

Fourth question: Since communication is already considered effective, no changes are needed for the next term.

Fifth question: Since communication is already considered effective, no changes are needed on the faculty coach’s end are needed.

*5. Has the team needed to interact with department staff personnel, i.e. the office staff or Kurt? Has this been handled in a professional manner? Were there any problems with these interactions?*

The only interaction the team has had with staff personnel is with Kurt for getting a VM. We initially needed a VM for our Jira instance, and it took a while to get access to it, but this was most likely because of all the requests in the beginning of the semester for other senior project teams.

*6. Does the team have a complete website with all project artifacts stored and up-to-date on the software engineering department webserver, i.e. linus.se.rit.edu? How often are entries on the webserver updated?*

As a group we need to do better with this aspect of the project. While our sponsor and faculty advisor have access to all of our artifacts as we have them on Google Drive and on GitHub, we had not updated them on the web server for quite some time, prior to the release 1 submission. The web server is up to date now with these artifacts.

*7. How well has the team made presentations to the sponsor and faculty coach? Was the interim project presentation done in a professional manner? What can be done to improve the team’s presentations?*

Our biggest challenge for our presentation was determining the best way to explain to system in a short amount of time that would be simple enough for everyone to follow. We feel that a flow chart in the presentation would’ve helped us been more effective in doing this, as our sponsor pointed out afterwards.

*8. How well has the team worked with other senior project teams, coordinating access to lab space and equipment, sharing experiences and ideas, etc.?*

Team Cobra has not had to coordinate access to any resources with other senior project teams.

### Achieving Customer Satisfaction

*1. In the team’s opinion has the work accomplished to date satisfied the project sponsor? Were there any weak spots in this regard?*

In the team’s opinion, based on feedback from the sponsor, the sponsor is more than happy with work accomplished thus far by Team Cobra. There were fears early on that we were moving on fairly slowly, but with the demonstration of our prototype at the interim presentation, the sponsor expressed surprise and satisfaction with how far we had progressed.