



Reflection

Cyclops Ride Assist: Real-Time Monitoring System

Team 9

Aaron Li (lia79)

Amos Cheung (cheuny2)

Amos Yu (yua25)

Brian Le (leb7)

Manny Lemos (lemosm1)

Table Of Contents

- [1. Revision History](#)
- [2. Changes in Response to Feedback](#)
 - [2.1. SRS and Hazard Analysis](#)
 - [2.2. Design and Design Documentation](#)
 - [2.3. VnV Plan and Report](#)
- [3. Design Iteration \(LO11\)](#)
- [4. Design Decisions \(LO12\)](#)
- [5. Economic Considerations \(LO23\)](#)
- [6. Reflection on Project Management \(LO24\)](#)
 - [6.1. How Does Your Project Management Compare to Your Development Plan](#)
 - [6.2. What Went Well?](#)
 - [6.3. What Went Wrong?](#)
 - [6.4. What Would You Do Differently Next Time?](#)

List of Tables

- [Table 1.1: Revision History](#)

List of Figures

1. Revision History

Table 1.1: Revision History

Date	Developer(s)	Change
2023-04-05	Aaron Li, Amos Cheung, Amos Yu, Brian Le, Manny Lemos	Document created

2. Changes in Response to Feedback

At the project's inception, we as a team created goals for what we wanted Cyclops Ride Assist (CRA) to look like, and created a development plan for how to bring these plans to fruition. Inevitably, as development marched on and as we received feedback from TAs and other teams, we realized that some goals needed to be dropped while others needed to be added. As our goals evolved, design decisions also changed.

2.1. SRS and Hazard Analysis

The [SRS](#) outlined the many functional and nonfunctional requirements for the project. One piece of advice that was given by the TAs was to include some sort of priority system for phasing in all the requirements. For example, to assign a higher priority to the requirements that are core functions of CRA, while assigning a lower priority for the requirements that can be addressed later or depend on the core ones. For example, we decided that creating a system that could mount on a bicycle (CNFR2) was a high priority requirement. Following this, creating a system that was compatible with all bicycle variants (CNFR14) was considered a lower priority requirement.

We also had to adjust the scope of the functional requirements in the SRS. One requirement we wanted to implement at the beginning of the school year was to have a forward facing headlamp that the user could turn on when cycling in dark conditions (CFR14). However, after testing, we eventually decided to remove the headlamp from our scope of functional requirements due to undue battery consumption and realizing that the requirement could already be sufficiently accomplished by already-existing solutions on the market. On the other hand, there were requirements we wanted to add that were initially not defined inside the SRS scope, such as a video/data capture button for the user (CFR15). It was added because we saw that the requirement would be a good complement to the video/data capture on crash identification (CFR8).

Amid these changes to our functional and nonfunctional requirements, we wanted to ensure that traceability between documents was maintained. Therefore, when requirements were removed, it did not rearrange the IDs of all the requirements in the list. Also, the requirements that were added mid-development were assigned new IDs, rather than replacing old IDs. This explains, for instance, the absence of the CFR5 or CFR6 requirements in the final draft of the SRS.

2.2. Design and Design Documentation

Our design had many iterations, as outlined in [Section 3](#) of this reflection document. It was necessary to follow an iterative approach to development rather than a "one-and-done" approach because it gave us the opportunity to identify issues in our design and revisit the drawing board.

A big change in our [design documentation](#) was in overhauling the layout. The document should be easy to follow, given that the objective is for an independent developer to be able to implement all the modules of our project without needing to consult us. These layout improvements were completed for the final draft. Furthermore, the TAs gave us feedback that our hardware specification was very well done, but that our software specification left room for interpretation. As a team of mainly mechatronics engineering students, this was helpful feedback and gave us insight into our strengths and weaknesses. For the final submission, we ensured that the software documentation in the MIS, as well as the commenting in our code files, was improved.

2.3. VnV Plan and Report

In our verification and validation process, there were some oversights that were pointed out to us by TAs and by other teams' peer reviews. We took the time to read over their suggestions and fix oversights where they existed in the both the [VnV Plan](#) and the [VnV Report](#). The first was the flawed logic of trying to test velocity data using an accelerometer. It was suggested to us that, if we really wanted to collect and verify velocity data, we should consider implementing GPS with a Kalman filter into our system. Deciding that this was outside of the scope of what we wanted to achieve, we decided to instead modify the accelerometer tests to focus solely on verifying acceleration data.

Another oversight that was pointed out to us was the absence of a usability survey. A usability survey is a great way for us to test and gather data about our product from the end users, that being cyclists. After receiving this feedback, we prepared a usability survey for our small group of clients to fill out. This usability survey is now defined as part of our VnV plan and included in our report.

Apart from these large changes, we had many small, incremental changes to our VnV documents, such as typo fixes, improvements in traceability, and in better describing the outcomes of the tests.

3. Design Iteration (LO11)

We did not arrive at our current design iteration by accident. It was a year-long process of gradual refinement from one prototype to the next. What the CRA is now is built on the shoulders of many poorer designs. And if the CRA continues development after this capstone course, we are sure that it will continue to get better and better.

Our first iteration was built on a simple breadboard. Components were wired together on the breadboard using jumper cables for quick prototyping. The CRA consisted of a Raspberry Pi, an ADXL345 accelerometer, an HC-SR04 ultrasonic sensor, some LEDs, and some loose cables that acted as buttons. Although the CRA was impractical at this stage, it was an important step toward a working embedded product.

Once we verified that the basic electrical and software components of the CRA were functional, it was time to transfer the breadboard onto a soldered circuit board. While we were soldering the components to the circuit board by hand, we were also using 3D modeling software to design a chassis for the CRA that we would eventually 3D print. This chassis consisted of a housing for all the electrical components, with a bike clamp attached under it. As we were assembling this iteration of the prototype for the rev0 presentation, we realized how difficult it was to put together and take apart. With this in mind, the housing would eventually be redesigned to align with a "design for assembly/manufacturability" philosophy. On the other hand, we got the bike clamp design correct the first time, and carried it over to the eventual final product.

This next iteration is the current iteration and the one we presented at the final demo. We decided to swap some electrical components around to improve the design. We swapped from using individual LEDs connected to the Raspberry Pi to a single 8-LED strip that we could purchase off the shelf. This served to dramatically simplify the electrical circuit design. We also considered the poor performance of the ultrasonic sensor we were using and decided to switch to a slightly more expensive, but much more reliable and performant LiDAR sensor. To simplify things further, we also swapped from using a soldered circuit board to directly wiring the connections to the Raspberry Pi.

The biggest change to the current iteration, however, was a complete redesign of the housing. Having learned from the first housing design, we changed the design to be more aesthetically appealing, faster to 3D print, and easier to assemble/disassemble. We tightened tolerances so all the components fit together snugly. We front-loaded our mechanical testing using 3D modeling simulations to reduce mistakes during 3D printing. We reduced the amount of unique components needed, and sped up assembly times by reducing the need to fiddle with small bolts and nuts.

4. Design Decisions (LO12)

Many important design decisions were made at the start of the project. Although, many decisions also changed as development proceeded, and as we, the team, learned more and more about the system. Many design decisions were already outlined above in the [Section 3](#), but we will continue to touch on important decisions that were made in this section as well.

One design decision that we carried from start to finish was our choice of mounting location for the CRA atop a user's bicycle. We began by considering all the available real estate on a typical bicycle, and landed on mounting the main housing on the handlebars, and mounting the rear vehicle sensor underneath the seat on the seat post. We also decided that, to connect the two housings together, we would use a water-resistant cable that would run along the frame of the bicycle next to the rear brake cable. We believe that this design decision was the optimal choice in terms of simplicity, installability, and usability.

However, some design decisions underwent an iterative process where multiple solutions were prototyped until the optimal design was found. The best example of this for the CRA was the sensor to use for rear vehicle detection. We began by using a camera paired with a computer vision algorithm to detect cars. This design was eventually scrapped because of the difficulty, complexity, and heavy resource usage of a reliable computer vision model. These issues were also pointed out by a TA during our rev0 demonstration. Thus, our next iteration used an ultrasonic sensor. It was readily available to us, simple to interface with, and inexpensive. It gave us a good idea of how distance sensors work, but was not performant. The ultrasonic sensor could only reliably detect distances up to 2 meters, which was insufficient for the use case of the CRA. We decided to scrap the ultrasonic sensor too. Our final iteration saw the Cyclops team elect to use LiDAR distance sensing. This methodology came with the improvements of significantly increased range, reduced noise when used in confined environments, and a more compact and durable design less prone to cable failure. By iteration, we arrived at our final design decision to use LiDAR as our rear vehicle sensor.

A controversial design decision noted by Dr. Smith in the revision 1 demo was the presence of separate power-on and power-off buttons. This decision was influenced by a constraint in the hardware chosen. The Raspberry Pi can be woken from a powered off state by shorting GPIO pin 3 to ground. However, GPIO pin 3 is the only I2C communication protocol SCL pin on the board. The adafruit accelerometer chosen for the CRA system communicates via I2C and uses the GPIO pin 3. As a result, the GPIO pin monitored for power-off cannot be the same GPIO pin used to power-on the Raspberry Pi.

5. Economic Considerations (LO23)

Our product is entering the biking accessories market, competing with other products that are designed to help increase cyclist awareness and safety during their biking trips. Marketing for our product would involve advertising and networking with retailers. Online advertising, especially advertising on social media platforms, would be a fantastic way to spread the word about our product and appeal to the young adult demographic, which comprises the majority of commuting cyclists. We could also reach out to bike retailers in Ontario to see if they would like to partner with us and supply CRA in their stores. There are many retailers like this, such as SportChek, Bushtukah, and Canadian Tire, so we are definitely entering a well-established market.

This current version of our product cost 400 CAD. This cost includes the prototyping process, such as components that we didn't end up using. If this product were to be mass produced, we could significantly reduce the cost. First of all, by buying all the components in bulk, we can reduce the cost per component. Secondly, we would switch from using a Raspberry Pi (which cost 250 USD, but proved extremely effective during our prototyping process) to a dedicated microchip. This would reduce resource usage, such as memory and computing power, and thus reduce cost. Thirdly, we would switch from 3D printing the housing to injection molding, which would accelerate manufacturing time and cost. After all is said and done, we estimate we could manufacture CRA at 100 CAD per product. If we independently sold each product at 120 CAD, we would have a profit of 20 CAD per product. This means we would have to sell 2,500 units of our product to recoup the [50,000 CAD cost](#) of starting up a mid-tier injection molding production run.

According to [Statistics Canada](#), an estimated 7 million Canadians report in 2013/14 to have cycled within the last 3 months. These numbers have increased since 1994/95 and are projected to continue increasing as long as Canada's population also continues to increase. These cyclists are all potential users of our product in Canada.

6. Reflection on Project Management (LO24)

6.1. How Does Your Project Management Compare to Your Development Plan

In the preliminary stages of the Cyclops project, rules and best practices regarding committing to the github repository were set into motion via the Development Plan. Github was expected to be used for all version control and collaboration. Furthermore Members were expected to maintain the repository by contributing clean and accurate code, and creating new branches when developing different aspects of software before merging into the main branch once functionality had been verified, approved, and was deemed stable. Additionally, commits made needed to have an associated issue number and issue that described what needed to be improved. Pull requests were to be processed with the validation of at least two other members within the team with discussions encouraged to be held within their respective topics. For the vast majority of commits to the Cyclops repository these guidelines were followed. This early decision and adherence to the [Development Plan](#) meant that our github repository has great traceability, organization, and maintainability. As a result, the Development plan was followed with respect to workflow.

Scheduled team meetings with an agenda and subsequent documentation proved to be an extremely fruitful venture that served the Cyclops team for the duration of the capstone lifecycle. Per the development plan, larger scope and big picture discussions, such as meeting minutes, should take place in the [Discussions tab](#) of the GitHub repository. These events enabled the Cyclops team to meet at regular intervals, discuss upcoming milestones, and bring our minds together to effectively determine the path we wanted our project to go down. The written agenda of these meetings worked to keep our team on track during meetings and cover a baseline amount of content. Moreover, these events allowed our team to distribute work evenly, and provide a voice for team members who wanted help. As a result, the Development Plan was followed with respect to communication and team meeting.

The technology outlined in the development plan has evolved significantly with each revision of the project. As technologies were implemented, tested, and deemed insufficient they were replaced with more formidable counterparts. An apt example of this is the evolution of rear vehicle distance sensing. The technology used for this function has evolved from a computer vision algorithm, to ultrasonic distance sensing, to the current state of LiDAR distance sensing. Evidently the original Development Plan was not not followed exactly, however its contents have been updated to reflect each revision of the product.

As the technology being used evolved, so too did the roles of different team members. The change from computer vision image recognition to ultrasonic distance sensing meant that significantly more time could be allotted to hardware design and testing. This saw one Cyclops team member shift their focus from software development to hardware housing design and software testing. Evidently the Development Plan was not followed exactly with regard to team member roles, but it it still quite accurate in its depiction of workload sharing and team members supporting one another.

6.2. What Went Well?

In the early stages of the project prior to the revision 0 demonstration, technological success was realized through the decision to purchase a Raspberry Pi development kit. This decision was significant because it began to form the confines within which our project must operate. This piece of hardware, which would become the backbone upon which the Cyclops project would be built, enabled the team to begin researching a range of hardware compatible with the IO provided, and design a housing that could contain and mount the boards footprint. Furthermore, the designed for development style of the Raspberry Pi aided tremendously by providing a hub to integrate, test, and optimize any new code.

As the project progressed significant problems were encountered. Initial goals of camera resolution and rear vehicle detection were not able to be realized. However, the Cyclops team rose to the occasion and took on these challenges with insight learned through this adversity. Prior to the purchase of the LiDAR distance sensor, LED strip, and power bank, tremendous research was conducted. A complete wiring schematics was generated, communication protocols and potential interference were researched in depth, power drawn from the Raspberry Pi and subsequently the power bank was analyzed. As a result of planning with the complete system in mind, the integration of these components was relatively seamless, and a final product that delivers the necessary specifications in a real world environment was achieved.

6.3. What Went Wrong?

When the Cyclops teams began to formulate an idea of what the final product would be, we had overarching goals in mind. Ideas of rear vehicle sensing, crash detection, and ride data logging were at the forefront of our minds. However, what was not considered as closely were the true specifications of the hardware that would be required to create a useful product that we would be proud to call our own.

Specifically, the front facing usb camera was purchased, software was written to drive video logging, and the hardware housing was designed around the camera. However, the potential output resolution of the camera was limited by the real-time video encoding constraints of the Raspberry Pi. This realization did not come until too late in the project when changing the implementation would be a catastrophic setback. If sufficient research was conducted into the bottlenecks of real-time video writing on the Raspberry Pi prior to the purchase of the camera, a Raspberry Pi camera would have been purchased which would enable hardware accelerated video encoding.

A significant source of time and money lost over the course of the project were the 3 iterations required to achieve accurate rear object distance sensing. Once again, in the early stages of the project a camera, and then subsequently a lidar distance sensor were purchased to achieve this functionality, however, the true requirements of this hardware was never considered in detail. The team only realized the glaring issues with the distance sensing accuracy after complete products were assembled and then tested.

6.4. What Would You Do Differently Next Time?

In response to the failures outlined above, a significant change in the approach to the early stages of project development would be put into action. When formulating ideas of what functionalities our final product would deliver, explicit requirements driven by real world experiences would be used. This could be achieved by taking to the streets with a vehicle, bike, and measuring device to determine at what distance a rider should be notified of approaching vehicles. Similarly, required camera resolution could be identified by using a cellphone to record a bicycle ride and determine what the minimum required resolution to effectively capture necessary details like license plates. With these driving requirements set about, a complete hardware lineup would then be researched that meets or exceeds the goals set about. However, this stage requires extreme analysis. The communication protocols, hardware interfacing issues, and libraries available to accelerate development must be carefully considered. The aim of this stage is to iron out any potential wrinkle which may occur during the integration process.

Following this preliminary phase another component of project development should be altered. A major point of time lost and frustration spawned from the several major integration events which occurred prior to demonstrations of our product. This consisted of the team bringing together all of our independent works and integrating them to form a functioning product that worked harmoniously. In future projects, the Cyclops

team has concluded that continuously integrating all of our works as we develop would lead to fewer roadblocks as the project progresses. Moreover, prior to any development of code, an architecture of the fully functioning software project should be constructed. From here the framework for module interaction, thread spawning and other implementation details can be determined. A top down approach to software design would once again alleviate significant pain points in the integration process.

While the Raspberry Pi brought significant success as a tool to integrate software and drive project development, it was also limiting in a number of ways. One major issue with the Raspberry Pi is its size. The hardware housing designed to hold the Raspberry Pi and all IO except the LiDAR sensor was quite large. This was a result of the footprint of the board itself being quite large. Furthermore, the Raspberry Pi has significant overhead from running linux. This results in poor battery life and a long amount of time to turn on and off. In the future the Cyclops team would use a development board that has the sole purpose of executing a single script in tandem with the Raspberry Pi development kit. This would serve to maintain the great development experience offered by the Raspberry Pi while alleviating the issues outlined.