Development of Robust Software Aboard the Coastal Profiling Float

Karl Ventayen

August 9, 2024

1 Abstract

Developing any observing system requires a robust design that can compensate for changes in technologies as well as the specific needs of the project. This is particularly important in the development of autonomous ocean observing systems which is the focus of my internship. MBARI has developed and is continuing to improve a profiling float targeting ocean health issues in the coastal zones and upper 350 meters of the world's ocean's, the coastal profiling float (CPF). The software engineering community has been developing robust design and development techniques for decades. This paper will discuss how I've integrated and adapted these techniques to the software I've written for CPF.

2 Introduction

There are numerous complex, critical, immediate challenges to diagnosing the problems affiliated with the health of the ocean. Among these problems is the NSF's 10 Big Ideas Problem: Navigating the New Arctic. As the very climates change landscapes internationally there is a need to understand various ecosystems such as the Arctic. Details such as the melting and

collapse of icebergs and glaciers are critial to understanding the impact of the changes that occur in the oceans. Devices and platforms such as the Coastal Profiling Float allow for us to gain a better understanding of the coasts such as those in the Arctic and other similar ecosystems.

MBARI has been developing the Costal Profiling Float (CPF) this is a device used to survey the ocean at various depths within the euphotic zones of the ocean (less than 350 meters in depth). This device is an in-water data aquisition platform of a larger ocean observing system. The data collection by the device is made possible by the sensors on the device. These sensors document details such as pH, Oxygen, Nitrate, Temperature, Salinity, Backscatter, Pressure (Depth), Florecense (Cholorphyll), and Optical Radiation (Sunlight). This device sinks to depths in the ocean off the coasts and as it rises it collects data from the sensors. Throughout the lifespan of this device it continues to collect detailed information in the waters around itself and relays it back to scientists for analysis. When all of the data is collected the information can then be extrapolated and interpreted as a profile along the route traveled by the CPF device. With this information scientists are able to more accurately diagnose the health of the ocean and the changes that occur over time with each profile.

3 Process Overview

Here is a decription of the process that I am learning and developing:

Understand the Problem: The composition of a problem can be sophisticated and comprehensive. Taking time to understand the problem allows to many details to be elaborated and smaller problems become apparent due to this abstraction of the material. This is especially useful when understanding how to approach various components that compose a problem or a massive problem is being analyzed for starting development or even further

development on currently existing code. In taking the time to understand and elaborate on the problem many key themes can be identified to work towards possible solutions, which can then be further elaborated later.

Do the Homework: In developing a solution it is appropriate to research possible ways to solve the problem. Since a problem can have many solutions it is best to decide the best one based on the circumstances and available resources such as time or money. In addition to understanding the problem it is necessary to understand there may be several possible solutions. Typically it is appropriate to understand there are many possible solutions for a single problem and the selected solution would be based on the circumstances at the given scenario.

Define the Specific Problem: The many problems that are associated with determining a solution, since it is rarely the case that a single solution with solve a whole problem. However they typically can not be all tackled at once. It takes may solutions to many problems to determine what would be an appropriate plan of approach to many of the problems to devise a solution. With all of the factors in mind it is possible to make an appropritate decicision, since it is valuable to allot time based on the priority of the selected problems within the bounds of a situation.

Develop a Conceptual System that meets a Defined Set of Requirements: When the proper background is applied to the problem as a whole it has context. This context with the research, problem, and possible solutions gives a person a better understanding what is possible with the given situtation.This background also makes the solution less abstract and allows for the user to understand the breadth of the situation, without convoluting the problem.

Iterate with Stakeholders: Stakeholders include, but are not limited to: project sponsors,

managers, and supervisors. When a design is being developed it is very appropriate to ensure that there is a way to make it possible to account for changes as the future stakeholders contribute to the project. These changes could happen for many reasons as well such as the needs of the project have changed, the project is being put in a different environment than originally designed to accomodate, or a change in the interest of the stakeholder. Therefore, there is a need to undertand the project as a whole in addition to the considerations acknowledged by the stakeholders.

Define a more fleshed out preliminary design that identifies risky elements and includes a preliminary design of the software and a test plan to prove the system works as everyone expects: When developing a process there are many aspects that need to be accounted for and needed to keep into consideration. One of the most significant being to acknowledge that any design is subject to change or further development. When a design is chosen there are many considerations such as risks have been identified and evaluated. There is no gurantee that there would be a definite solution to the problem, however many of the known unknowns can be handled. Known unkowns are known problems that have the possiblity of occuring, such as the CPF washing to shore and the depth sensor reading values less than zero since the device is above sea level. But in contrast unknown unknowns are more difficult to understand and account for since they are less predictable. An example of an unknown unkonwn is a value in the system gets corrupted and the read values are inconsistent. This is where these cases are to accomodate for the unexpected. The follow up question being: How is the device then to behave in response these unexpected behaviors?

Start Coding:

Using a combination of established and personal software design guidelines. This includes decisions that have been developed using techniques within the realm of developing such as

decisions for version control, integrated development environments, choice of language, etc. Many of these details have been adopted from the Chemical Sensors Lab (MBARI). Some ideas used are as described below:

SOLID Design Principles: Principles such as the Single Responsibility Principle which refers to functions having a single purpose and serving that purpose alone. When debugging a program it is appropriate to follow this guide to avoid confusion as more functionality is added to the application overall, since the interface is simple is more straightforward to understand.

MISRA C: These are a series of guidelines designed for critical systems. Systems like the Coastal Profiling Float are critical systems.

Unit Testing: Focusing on unit testing straegies is an important detail to gain from this. Having unit tests allows for the further development on systems based on the expected input and moving forward based on the known values. The comprehensive the tests for the program are the more robust it would be.

Documented Flow of Information: When the flow of information makes sense it is more apparent how the functions work and where to look in regards to debugging programs with many dependencies.

4 Anticipated Results

Through the development of the communication systems on the coastal profiling float a method for development is expected to emerge. Processes to understand the problem and define the primary characteristics that resolution are crucial to understanding what is expected of a viable solution. Next, research allows for one to gain a background appropriate for understanding the problem and possible solutions. Through having thorough understanding

of the context of the problem it is possible to make more informed decisions in regards to resolving the problem. Additionally, designing a solution before implementation allows for an evaluation process to analyze possible ways to solve the problem. Once completed the best course of action can be determined. Finally, texhniques sych as unit testing allow for code to be tested thoroughly. Since no solution is perfect constant testing allows for problems such as edge cases to become more apparent. When systems are developed, such as those on the coastal profiling float it is expected that a well thought methodology is devised.

5 Metrics of Success

When code is properly analyzed, tested, and simulated. When analyzed the naming conventions, formatting, and styles are observed and evaluated. The importance of these observations is determing how understandable the code is written (human readability). When these methods are implemented appropriately it is better understood by other people and they would be able to develop the code further.

Furthermore, effective conventions are able to accomodate for new requirements as needs arise. Since project requirements constantly change, especially long term projects such as the coastal profiling float, there is the need to acknowledge changes that may occur. The applications of this allows for a robust system that can last for a long time.

Additionally, code should be maintained by the appropriate level of future staff. As time moves forwared it is expected for team members to rotate as well. A robust system for producing code should permit for future development team members to understand the development process from intern to entry level to senior developers.

As the changes occur the system should scale as the application needs change. Due to

the lack of predictability of a development team's finances and employment it is appropriate to understand that the scale of the application may change ahd the attention to certain aspects of the project may vary through time. A robust system permits for these scaling changes to take place.

6 Broader Impacts

Having a robust method allows for accessibility among those who need to understand code and further develop existing code. This includes entry level developers as well as scientists and researchers. In software engineering it is typical to start development immediately after employment. When a robust system is devised the onboarding process is easier for the new engineers to learn the current state of a project and for a mentor to train the new employee. Additionally, in terms of the coastal profiling float a robust system allows for others such as researchers and scientists to make changes to code and adapt it to their needs. When code is robust it is accessible and that allows for many people to be able to learn about how it works and make changes as needed without being a professional in the field of Computer Science. In scientific exploration this is found to be helpful in cases such as when a computer scientist is not on a scientific team. Robust systems grant people the ability to learn and gain programming skill sets to further existing code and adapt it for their personal needs.

7 Data

7.1 First Test Input to Program

This is the output for the input: `$testing, 123, 234, 345, 456`

```
1 message: $testing,123,234,345,456
2
3 testing,123,234,345,456
4   -----FIRST IMPLEMENTATION-----
```

```
  Executed: $testing
  Parameters: 123 234 345 456 0
  Runtime: 0.01 seconds
  -----SECOND IMPLEMENTATION-----
  Executed: $testing
  Parameters: 123 234 345 456 0
  Runtime: 0.003 seconds
```

7.2 Second Test Input to Program

This is the output for the input: `$tasting, 123, 234, 345, 456`

```
message: $tasting,123,234,345,456

tasting,123,234,345,456
  -----FIRST IMPLEMENTATION-----
  ERROR - Invalid command: tasting
  Parameters: 123 234 345 456 0
  Runtime: 0.011 seconds
  -----SECOND IMPLEMENTATION-----
  ERROR - Invalid command: tasting
  Parameters: 123 234 345 456 0
  Runtime: 0.004 seconds
```

8 Discussion

As expressed in the example program there are expected outputs from both of the implementations. By observation there are two implementations and they both have different run times. The first implementation runs slower than the other implementation. More specifically, this occurs due to the usage of available libraries. Additionally, both of the programs were able to register the command and parameters. In the first example the first command

is registered as `$testing` and the parameters are recognized as `123, 234, 345, 456`.

However there is a problem that is recognized in the second program being run. In the second program there is a problem in the command itself instead of `$testing` the value `$tasting` is inputted, which has an effect on the expected output, yeilding the following error: `ERROR - Invalid command: tasting`. This error recognizes the problem in the command and displays an error to the programmer. This an important characteristic to be able to recognize in the program because many cases need to be handled and accounted for.

9 Conclusion

In conclusion, we need to be able to understand processes to be able to effectively develop systems and ensure quality of programming. When implemented appropriately it is possible understand what makes the program work or not work. And when tested it is possible to ensure code to function as a whole.

This is a comprehensive process that allows for understanding project needs to testing the implementation for a solution. Through a rigorus process I was able to understand how to ensure that may of the process are accounted for and make accomodations for the needs as they arise.

10 Works Cited

7 ways to make embedded software safe and secure. www.mathworks.com. (2019). https://www.mathworks
team/Objects/w/93170v00_7_Ways_Whitepaper.pdf

Villanueva, N. (2023, February 6). Solid design principles by Uncle Bob explained. DNAMIC.

https://dnamic.ai/blog/introduction-of-solid-design-principles-by-uncle-bob/