12-1-2021

THE VISUALIZATION OF WIRELESS NETWORK DATA IN AUGMENTED REALITY

Tiffany Frias

THE VISUALIZATION OF WIRELESS NETWORK DATA IN AUGMENTED REALITY

by

TIFFANY FRIAS

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2021

ACKNOWLEDGMENTS

I'd like to acknowledge my software team, WiDROS, where with many challenges never gave up showing results. The team consisted of computer science majors Daniel Tam, Joell Soriano, Renato Alex Cruel Amado, Joshua Pearson, and me, Tiffany Frias. The team always put forward their best skills to keep the project moving.

I would also like to acknowledge my faculty mentor, Dr. Chris Conly, for always being available to help and being accommodating to the team through our project challenges. I would also like to acknowledge Jesse Lee, our sponsor from Raytheon Technologies. The team met with him every week to discuss project progress for his feedback on the state of the project.

Lastly, I'd like to thank my family that set my expectations high to have desired to graduate with an Honors College degree. Not only that, but they also allowed me the opportunity to attend university as a first-generation college student to study the field of computer science and become the first software engineer of the family. My mother had been the first person to ignite the idea inside me to earn a degree in computer science, which has now defined my academic and professional career forever. I hope to inspire other computer scientists within my family and people around me so they can learn and contribute to the challenging technical complexities.

November 29, 2021

ABSTRACT

THE VISUALIZATION OF WIRELESS NETWORK

DATA IN AUGMETNED REALITY

Tiffany Frias, B.S. Computer Science

The University of Texas at Arlington, 2021

Faculty Mentor: Dr. Chris Conly

The visualization of wireless networks has been a research topic so computer scientists can understand what cannot be seen with the naked eye and advance in specialized fields such as data visualization, cyber security, and computer networking. Flying a drone with a microcontroller around a building, signal strength and location data points of wireless networks can be collected and uploaded into a cloud database like AWS (Amazon Web Services), or Microsoft Azure Cosmos DB. On the Unity platform used for game development, developers can create a program that pulls such data from the cloud in real-time.

Using this data, they then map the location of the wireless networks to the program world and use the strength to adjust the transparency of the visual shape that represents the wireless network. This program is run on the Hololens 2, so users can see the shapes that

represent the wireless network data first-hand when looking at the exact building where the wireless networks are being detected.

However, projects that aim to achieve end-to-end data transfer goals like these are subject to obstacles that require adaptions to software and environmental constraints, which require agile software development methodologies for constant modification of project expectations for client satisfaction.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION: PROJECT OVERVIEW

1.1 Project Charter

To understand the purpose of the Honors aspect of the senior design project, it is significant to discuss the background and overview of the senior design project for the College of Engineering. The problem statement for this project was that the client had the desire to scan access point wireless network signals and indicate their location and ranges. Such valuable information is then used for defense projects, logistics, cyber security, and research in computer networking. A straightforward method to visualize wireless networks highly increases productivity and innovation when it comes to researching what cannot be seen with the naked eye.

1.1.1 Background

The company sponsor's use case for this project was to be able to monitor access points of wireless networks such that appropriate responses and actions are performed towards the signals that are being received. The company believes that such signals assist in providing information on the most updated status about the access points being monitored. Dead spots and optimal access point placements are monitored with such a system. Other use cases that the company sponsor required were to visualize WiFi networks as domes in augmented reality through Hololens 2, the augmented reality glasses created by Microsoft. Such a program would be utilized to analyze one specific region's wireless network coverage. Raytheon has been a project sponsor for the College of

Engineering for several years and had requested the team to work on this project to improve their monitoring systems. Existing project requirements and work from previous teams were to be connected to the current team to continue working on the assigned project.

1.1.2 Methodology

The team was to create an access point scanning script on the Raspberry Pi, the microcontroller that can detect wireless networks. All data is collected and put into a JSON file for a database to collect and store. From there, the team can pull the data from the database for analyzation. The JSON file is uploaded to the database infrastructure that was built off an SQL network. It was capable of hosting Application Programming Interfaces, also known as APIs, which retrieve the required data to map the data points in the Hololens 2. In other words, the team is to collect the data coming from the SQL database and display it to the user in augmented reality.

1.1.3 Related Work

FEA, or finite element analysis, is a computerized method that can predict how a certain element will react to physical forces such as heat, vibrations, or fluid flow. Structural analysis of elements can be done prior to constructing their concept. FEA is usually only done on computer-generated graphics, however, three engineers developed an integrated platform to be able to perform real-time FEA simulations in augmented reality (Huang, Ong, & Nee 2015). They discussed how an AR system typically has three fundamental modules, which are tracking, rendering, and interaction. The WiDROS team took from this source that the first steps into starting the project was choosing the method of how to track the wireless networks, how to render them into the Unity program, and the way in which the user was to interact with the environment.

Another related work was done by students in a telecommunication science laboratory in Maryland, USA where they discussed the ways such that the Microsoft Hololens could assist human operators perform network operation tasks. They developed three applications, the first being a three-dimensional network visualizer that was able to distinctively display network topologies in two levels of detail: a global view and local network view. From the global view, the application can simulate alerts when the user's attention is required. Once the user chooses to address the alert, their program switches from a global to a local view of the region of wireless networks (Beitzel, Dykstra, Toliver, & Youzwak 2017).

That application had similar project objectives assigned to WiDROS. Along with having to detect wireless networks, it also utilized the Unity game engine as the core development platform. However, this project was different in a way such that the WiDROS team was assigned to only pick up wireless network data from a building at the University of Texas at Arlington and the program will visualize the wireless domes right on the building, so the user physically must be standing in front of the building at the University of Texas at Arlington. Their application showed all active networks on a visual of a globe and the user was to interact with a certain area for the local network regions to expand.

1.2 System Requirement Specification

The WiDROS team had to follow strict requirements for the project due to the sponsorship with Raytheon Technologies. Due to privacy concerns, the first functional requirement was to keep the code and project information private, so public repositories on Github or the public sharing of the project details were not allowed. After the restrictions were set, the team was given the following requirements for the system:

- The system shall collect wireless network data to upload to a database.
- The system shall display WiFi signals as domes and their respective metadata.
- The system shall display an AR environment outside the Engineering Research Building at the University of Texas at Arlington.
- The AR environment shall be mapped to GPS coordinates of the real-world campus environment and display the wireless network signals with varying transparencies according to the relative strength.

The following user requirements were also assigned:

- The user shall be able to see where a wireless network is detected in the AR environment.
- The user shall be able to click on signals to learn more about that wireless network.
- The user shall be able to maneuver around the AR environment.

Hardware logistics were also defined, such as the system requiring an AR headset, and the AR environment being able to run on a Windows 10 PC. Performance requirements were defined, such as being able to update relevant portions to the data file of the wireless network domes every half minute while the Raspberry Pi is collecting data. This is due to the fact that wireless networks could decrease in strength or no longer be available.

1.3 Architectural & Detailed Design Specification

Understanding the architectural integrity of each subsystem that takes part in the project was significant to understand how the data is being collected from the previous subsystem, understood, processed in the current state, and how it is sent to the next subsystem. Going into such depth of the design specifications required a separate document, so the following is a summarized version of the team's architectural and detailed design specification.

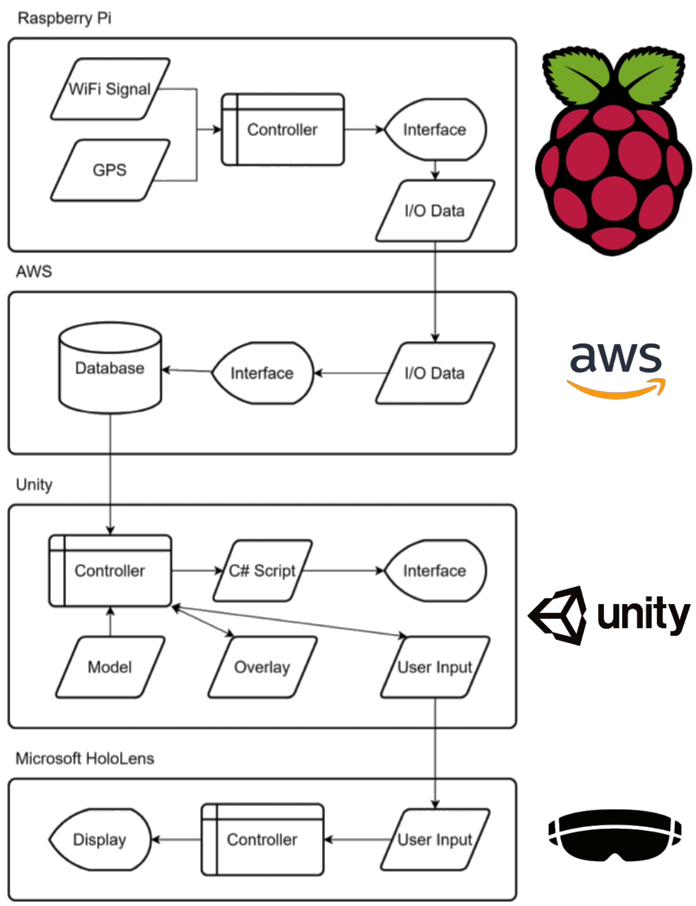

Figure 1.1: End-to-End Data Transfer Starting from the Raspberry Pi to Hololens 2

The initial architectural system was as described in the above diagram. First, wireless network signals and GPS coordinates were collected by the controller of the Pi, sent to the interface, then the controller sent this data onto a cloud database like AWS. The data would go through the database's interface and be stored in a real-time database container. However, AWS was not the final database service selected to utilize, which will be explained in Chapter 4. From there, the game engine platform, Unity, obtains the wireless network information through a C# script in a project program that interacts with the cloud database. This script pulls the contents from the database and allocates the data to the proper variables for development. From there, every wireless network is considered as well as their metadata and location data, then mapped as spheres on the program to be visualized as domes in an augmented reality environment.

Now that the senior design project for the College of Engineering has been described into its detail of background, methodology, related works, requirements, and design specifications, the Honors aspect that is attached to this project can now be explained to refined detail. Results and work of the Honors aspect of the project will be shown. However, the results and state of the project overall will be discussed in Chapter 5.

CHAPTER 2

HONORS REQUIREMENTS

2.1 Data Visualization

There were two components set to complete as part of the Honors degree, the first being to work on the best way to visualize the wireless networks on the Hololens 2 and how users would go about interacting in the environment to learn more information about the network: the data visualization. After thought and creating proofs of concept, the proposal idea was brought up to the Raytheon sponsor for approval to proceed with the project idea. The second component was being a product owner for the team, which came with heavy responsibility.

2.1.1 Information Display to Hololens 2 User

The first problem to solve was how to show the user the metadata of wireless networks without bombarding them with all the information at once. Especially in an environment where there are potentially hundreds of wireless networks being analyzed, showing every single detail of every single network is not ideal. Also, some metadata is irrelevant to the user or can already be visualized in a way such that the text does not necessarily have to be displayed to the user. For example, telling the user that a wireless network was detected at the point of (4.5m, 6.2m, 7.8m) from their position, assuming they are at the origin, is not necessary if you just map the visual to be at that distance in the Hololens environment. However, information like SSID (wireless network name) does make sense to display through text

Prior to completing the end-to-end data transfer, the data visualization portion of the project had already been accomplished. The first was a program that visualized a basic cube in the ceiling that signified a wireless network. Upon the user going through the program using the Hololens 2, they can see the cube in view.

Figure 2.1: Observed Cube in Hololens 2

It is not until the Hololens 2 picks up that the eye is looking right at the cube, the metadata is then shown immediately to the user.

Figure 2.2: Eye-Tracked Cube in Hololens 2

Upon demonstrating this proof of concept, the company sponsor approved the method of metadata visualization but proposed that an area for improvement was to add strength numbers in the information display.

2.1.2 Wireless Network Visualization

It has been common in other related works to pick up wireless network detection with heat maps, or, like in the related work from Chapter 1, the networks were presented to the user at a smaller scale as shown below.

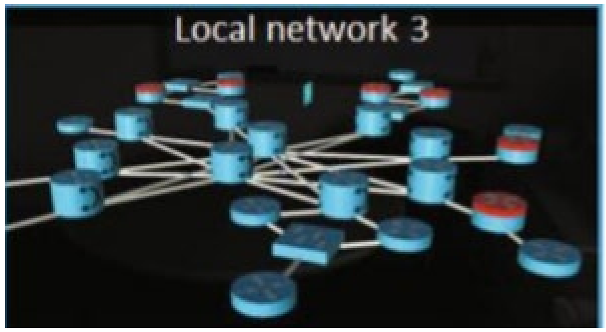

Figure 2.3: Network Visualization by Beitzel, Dykstra, Toliver, and Youzwak

Since the initial project requirement was for the user to see the networks on the building, the scalability had to be set for real-world sizing and simulation of the wireless network data collected. The Honors portion then presented to the sponsors was to create a dome for every access point such that the size will be denoted based on the signal range of the access point, with the origin of the dome being the physical location of that access point.

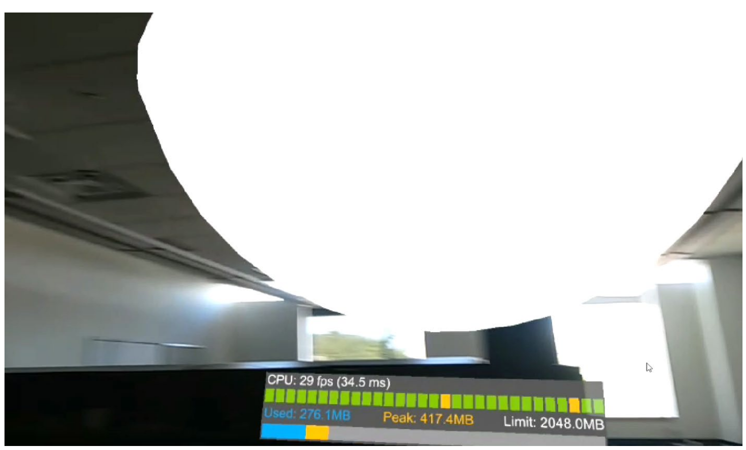

Figure 2.4: Wireless Network Dome

For further refinement, the higher the strength of a network, the more opaque the shape was to be. In the case the strength was low, it would appear transparent on the sphere. Usually, this would mean that the closer to the origin, the stronger the network, therefore the more opaque a sphere was at its origin.

Another issue considered was, with all the domes being the same color, how bad would the overlapping be? Because of this, a third project was done to visualize the networks in different colors. This time, it was performed outside the Engineering Research Building at the University of Texas at Arlington.

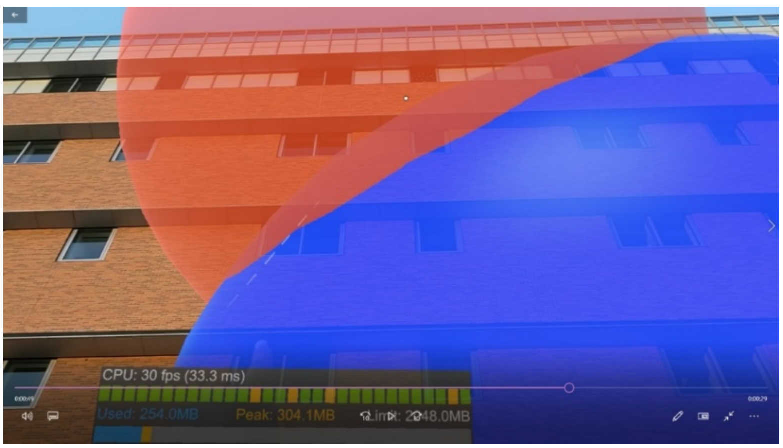

Figure 2.5: Colored Network Domes Outside ERB

Using different colors helped visualize when two networks were overlapping yet distinctively independent from each other. This way the final data visualization was to randomize color schemes across all wireless networks in the case of overlapping. In further research or continuation of the project, there could be a flag in the program to check if two overlapping wireless networks were assigned the same color. If so, then the program must assign one of the bubbles a new color.

2.2 Product Owner Responsibilities

As product owner of the WiDROS Fall 2021 team, the product owner was already assigned to take care of all external communications and taking on the initiative for the team to gather. Over the first semester of Senior Design I, the product owner held responsibility for communicating with the professor about the project information status given by the company sponsor. The product owner kept communication flowing between the team, the professor, and company sponsor.

In the second semester, the product owner oversaw requesting the proper technologies for the project to Dr. Conly or, in the case of the drone, asking Dr. Wetz. Every week, the product owner met with the project sponsor to discuss what current tasks were accomplished that week, the feedback on said tasks, what the tasks were for the following week, and then a quick overview of the project flow throughout the whole semester. Given this feedback, the product owner prioritized what tasks had to be done the following week and was responsible for measuring performance of the team.

CHAPTER 3

AGILE SOFTWARE METHODOLOGY

3.1 The Waterfall Process

Prior to the 1990's, software processes were held under solid requirements at the start of the project and development was executed in strict sequential steps. The process started by establishing requirements, analyzing costs, designing the system, implementation, and testing. Finally, the project would deploy, and maintenance was kept if the product was successful. None of the steps above overlapped and were difficult to modify because of it. The process just described is the waterfall model, a linear approach to the software development lifecycle. When flaws are found, the process requires starting over entirely. Error correction is not considered, and all testing is delayed until the end of the cycle (Lewis 2019).

This traditional model disregards the fact that errors are bound to occur in projects. In industry, management found themselves in trouble for following strict requirements assigned by the customer at the start, and only shown the final product once the process was over. Realistically, software clients tend to not know the exact requirements they need. Because of the many disadvantages to the software development process in the middle of an era where technology advances are strongly desired and capable, a new software process was adapted.

3.2 The Agile Software Process

The agile software process highly values the customer's needs and has constant requirement changes and client interactions to meet that. Under the agile manifesto, the agile manifesto authors state they are for customer collaboration over contract negotiation, responding to change over following a plan, working software over comprehensive documentation, and individuals and interactions over processes and tools (Beck 2021). Such methodology has been extremely successful in the industry. Statistically, development teams under an agile development process have 64% success rates, while teams under the waterfall model are 49% (Parody 2018).

There are many advantages to using the agile software process over the waterfall methodology. Changes are allowed after requirements are made such that they meet the client's satisfaction. This means that the client is actively involved and aware of the project progress and outcomes. In addition, it is natural for errors to occur in projects. When it comes to an agile methodology, such errors are expected so it is easy to identify and fix them without repeating the whole process.

This methodology was ideal for the WiDROS team because there were a lot of changes in project requirements. It was vital for the company sponsor to be content with the project outcomes since they requested the product and invested funding into the tools, such as the Hololens 2, the cloud services, and other miscellaneous technologies or tools.

3.2.1 Agile Scrum Teams

WiDROS was an agile scrum team, meaning the team was susceptible to sudden and constant project changes. This will cover what the agile software development process is, what agile scrum teams do, and the responsibility and significance of a product owner in an agile scrum team.

The WiDROS team implemented an agile scrum team with a small twist. A scrum team is a team with a scrum master, product owner, and a development team of usually five to nine software developers that focuses on completing the required project deliverables while keeping constant communication with each other such that they all share the common goal to finish objectives in the same convention.

Every two weeks, a set of tasks or goals are written out in task-keeping software, like Jira, such that the team can organize the task assignments and track progress. During these two weeks, called a sprint, the team aims to accomplish these objectives and they have a meeting every day, hosted by the scrum master, to each discuss what they worked on the day prior and their tasks to accomplish today. The scrum master then asks if anyone is going through any roadblocks or mitigations in their work such that they can give a nudge to the other teams that some tasks may be dependent on.

Another important role in an agile scrum team is the product owner. This role is the direct communication link to the customer and business partners, and it is their responsibility to thoroughly understand the customer desire in the product in development. They are also always available for the team to communicate with to ask about the specifics to their work requirements. Lastly, they manage the product backlog. This is the set of tasks that need to be completed for the entire project that are prioritized and selected throughout the sprints. They describe each task's detail to the highest possible extent such that the developers can execute them correctly.

3.2.2 The Agile Scrum Changes of WiDROS

The WiDROS team had to perform the agile scum methodology with a few modifications. First and foremost, the team did not always meet every day. As students, this project was not the primary and solitary focus of the team the way it would be had it been in industry. Secondly, there are only five members in the WiDROS team. This required the product owner and scrum master to engage in completing the development tasks, and the product owner prioritized the tasks to complete every week. In the case of mitigations, it is the product owner instead of the scrum master who talked to either the senior design professor or the company sponsor about the mitigation and what needed to be done to overcome it.

Even with the modifications to the traditional agile scrum conventions, the team benefitted greatly from the advantages, like being able to communicate with the customer frequently and errors being easily detectible and addressed in a span of a sprint. However, the constant requirement changes to the project brought the team many obstacles which will be discussed in the next chapter.

CHAPTER 4

PROJECT OBSTACLES & RESULTS

This chapter will cover all the struggles and obstacles that not only the product owner faced and fell responsible, but the challenges faced by the team to adapt to the fluid nature of the project. The team quickly learned to adapt to new project requirement modifications as well as working through the learning curve of new technologies such as AWS (Amazon Web Services), Microsoft Azure, and the Unity game development application. Even when the technology in question was no longer part of the project, the team transferred the skills learned onto the new technology instructed to use, leading to the success of overcoming the obstacles.

4.1 Complete Change in Project

During the first semester of senior design, the WiDROS team was part of the Sensorium team. Sensorium was an ongoing electrical engineering project also sponsored by Raytheon to create a super sensor with twelve sensors that uses machine learning to recognize the occurrence of events in a controlled environment. The WiDROS team are all computer science majors so they were assigned to work on the software components of this project. The electrical engineering team lead helped to familiarize the team with the Sensorium code. However, within the first two weeks of the second semester of senior design, Raytheon scrapped the Sensorium project. The electrical engineering team was assigned to another project, and the software team was assigned to what is now WiDROS.

From there, the WiDROS team already had a setback in the fact they were unaware of their project until the second semester of senior design. After receiving the project assignment and requirements, it took a few weeks to collect the technologies, such as the Raspberry Pi, a drone, and Hololens 2.

4.1.1 Weekly Changes & Challenges

After the entire project change, there were still ongoing changes to the WiDROS project due to the fact it was being developed under the agile software development process. The sponsor at first requested a static implementation of visualizing wireless networks, which would not have required implementation of a cloud database. The Raspberry Pi could have easily sent the data to a text file for the Unity program to load and statically map the data domes as read in, without concern as to whether any changes had occurred, or were occurring as it ran. The next week, the sponsor thought it over and discussed how they now preferred a dynamic implementation instead, due to the fact they wanted to see the details about the networks in real-time.

While such requirements were being switched back and forth, the team found it difficult to focus on what to do owing to the fact that the project requirements could have reverted back to static. There would be no purpose in learning about the cloud services like Microsoft Azure or AWS since a static implementation would not involve either one. The initial requirement had also been to use AWS as the cloud service for the dynamic implementation. Since no one in the team had AWS experience, the team was watching online tutorials on AWS regarding setting up the database, how to upload data through the script, and how to access such data from a C# script on the Unity platform.

Halfway through the semester, the WiDROS product owner received word from the lead of the former Sensorium EE team that the Department of Electrical Engineering could no longer keep paying for AWS due to the scrapping of Sensorium. This shifted the team's priority from learning about AWS to then finding a new cloud database that was affordable and suitable for the project requirements.

After hunting for affordable cloud services, the product owner decided on Microsoft Azure Cosmos DB, a database free for students to use as long as they do not exceed the data requirements. However, the project deliverables were going to be due in a matter of weeks, so the team set Azure learning on hold.

4.1.2 Solutions to Success

While the project requirements were unstable, it was important for the team to continue progress where possible to ensure the deadlines of the deliverables were met. While the sponsor went back and forth on the implementation, the product owner prioritized data visualization so that the team had a proof of concept to show while the data transfer method was being discussed. Because of this, by the time the end-to-end data transfer was complete, all the team had to do was use the programs created for proofs of concept that had the domes and set the positions and strengths according to the data.

Connecting the database the team created on Microsoft Azure was put on hold due to the difficulties of uploading and connecting to the server on both ends of the Pi and the Unity application. However, the team wanted to complete the end-to-end data transfer regardless such that they could show that the mapping of the wireless network domes was possible. As a result, the team created their own homegrown SQL database that takes the JSON file from the data collecting script. From there, the C# script used to collect the data on Unity was much more facilitative than both AWS and Azure. Providing quick solutions to the sudden changes in project requirements were what kept the team going to success.

4.2 Project Changes in Industry

The WiDROS team experienced many project changes, but the team should expect these challenges throughout their entire software career. As discussed in Chapter 3, agile teams go through constant requirement modifications, but it is worth the changes for the customer to be pleased with the final product. That is why as a software engineer, you must be adaptive and a quick learner, and as a product owner you must be that and more. A product owner and other management roles in a team have to make important decisions throughout project processes and the outcomes always fall to their responsibility.

According to the Standish Report in 2014, about 30% of projects are failures in industry. These are the projects that are scrapped due to multiple factors, such as budget or demand. More than half of industry projects go through some feature change. Now that the agile methodology has been adapted by most industries, it comes to no surprise that industry projects make significant shifts in the process. Lastly, more than 83.8% of projects in industry are either over-time or over-budget (Fu 2017).

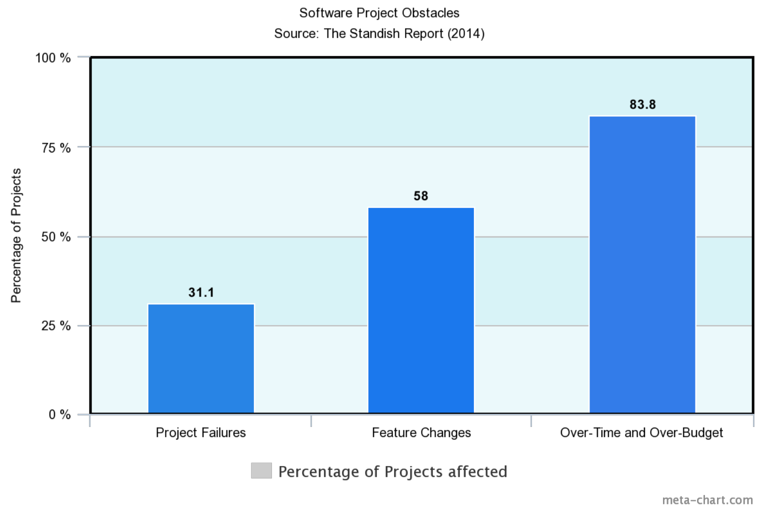

Table 4.1: Software Project Obstacles

There are roles specialized to analyze the cost and timing of projects, but they are never going to perfectly predict the project outcomes as they depend on both the software resources and the client's approval. That is why it is important for companies to evaluate the risks in the case of worst-case scenarios of time and budgeting, and software teams must be adaptive and as productive as possible.

CHAPTER 5

CONCLUSION

The WiDROS project was about flying a drone with a Raspberry Pi attached around a UTA building such that it collects wireless network data for a user to view on a Hololens as domes coming out of that same building. There were two Honors aspects assigned to this project. The first was to focus on creating the best method to represent the data to the user and allow for user interaction when they want to learn more about a network. The second Honors aspect was being the product owner of the WiDROS team, which firstly meant being in charge of the project features, goals, and outcome. Secondly, it meant keeping track of all the tasks to execute and prioritize what was going to be the team objective for all eight sprints. Lastly, it was being the prime communicator among business stakeholders, which in the case of senior design, was the company sponsor and the senior design professor.

5.1 Project Final Results

The Raspberry Pi successfully detected wireless networks and collected their information, such as the name, signal strength, MAC address, and signal channel. The GPS coordinates for the general area of network origins were collected with the help of a GPS specialized for the Raspberry Pi with an antenna to help it detect networks indoors. A proof of concept was created of the wireless network domes on the Hololens. Most importantly, the end-to-end data transfer from the Raspberry Pi to the Hololens 2 was successful.

5.2 Lessons Learned

A significant lesson that was learned from the project was to be adaptable and a quick learner. It is common in industry for software projects, especially in agile processes, to take significant shifts in direction and perhaps do not require technologies that were being used prior to the shift. A lot of new technologies and skills were involved, like developing Unity programs, setting up an AWS and Microsoft Azure database, and how to program for the Hololens 2. Perhaps such skills are not needed for the roles the team take on in their academic career. However, it is the skill of being able to pick up new tools and quickly learn how to use them that will be taken from this project and applied for the rest of their academic and professional careers.

A last lesson learned as someone trying to step into management was to take initiative for responsibility and motivate the team in times when there are obstacles, whether the projects are going as planned or are going through challenges. Without someone leading the way, whether it is a product owner, or the project client, direction and priority of tasks need to be established for the team such that they are productive and execute the results desired.

REFERENCES

Beck, K. Beedle, M. Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Martin, R., Mellor, S., Thomas, D., Greening, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Schwaber, K., Sutherland., J. (2021) Manifesto for Agile Software Development. Agile Alliance

Beitzel, S., Dykstra, J., Toliver, P., Youzwak, J. (2017). Exploring 3D Cybersecurity Visualization with the Microsoft Hololens. Vencore Labs, Basking Ridge, NJ, USA.

Huang, J. M., Ong, S. K., & Nee, A. Y. C. (2015). Real-time finite element structural analysis in augmented reality. Advances in Engineering Software, 87, 43–56.

Lewis, S. (2019). waterfall model. SearchSoftwareQuality.

Parody, L. (2018). How to Manage Modern Software Projects: Waterfall vs. Agile. Medium

BIOGRAPHICAL INFORMATION

Tiffany Frias is a computer science major with a minor in mathematics. Throughout her academic career, she has focused on technical complexities, such as theoretical computer science and algorithms, computational numerical analysis, and working in agile software teams.

For the Honors College, she worked on projects in physics, linear algebra, theoretical computer science, computer networking, and software testing. The largest project was computer networking, where she first developed a client-server script for a graphical user interface to allow client communication. The second component was writing a research paper on the technical concepts of Bitcoin to present to the professor, who had an extensive background in blockchain research.

Some of the most interesting projects across her academic career have been creating a group-based movie recommendation model, developing an Android note-taking app as part of an agile software team, and taking part in project Slash-Trash, a database project created to motivate users to bring reusable items to restaurants for rewards.

Upon graduation, Tiffany is to begin her career as a software engineer in an agile scrum team at Charles Schwab Co., where she interned prior to her last undergraduate semester. That summer, she was also inspired by directors and upper management she met to one day earn a master’s degree in project or business management so that she could lead a software team in industry.