5-2024

Software Engineering Internship Portfolio

Shawn Eidem
St. Cloud State University

Software Engineering Internship Portfolio

by

Shawn L. Eidem

A Portfolio

Submitted to the Graduate Faculty of

Saint Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of Professional Master of Science

in Software Engineering

May 2024

Portfolio Committee:
Maninder Singh, Advisor
Omar Al-Azzam
Adriano Cavalcanti

Abstract

This document details the experiences and knowledge gained during the internship capstone course of the PSM degree in Software Engineering. The capstone was split across two separate internships at TSI Inc. and MiTek. My role with TSI was to support global production activities involving supporting legacy software, developing new software for production output improvements, and software validation. Additionally, this role involved collaboration with various trade backgrounds, global troubleshooting, database administration, and project planning. Over the course of the internship with TSI the main focus projects to highlight and discuss were CFlow, Sensor Database Annealing System, and Calibration Time Analyzer. The second half of the portfolio with MiTek discusses work done primarily on the Array software as part of a agile-scrum oriented team.

In this defense, I detail my internship responsibilities, insights gained, skills developed, and practices learned. I discuss the impact and benefits of the highlight projects as part of my internship. Additionally, I discuss a supplementary project created as an addendum to the internship experience that highlights design and technical skills acquired throughout the internships. The report concludes with a reflection on internship benefits and closing thoughts.

Acknowledgements

I would like to take this opportunity to thank my family and friends for their pillar of support throughout the process of obtaining my degree. Without them, I would be morally dejected and may have given up before making it to this point. Additionally, I would like to thank my colleagues for providing insight, guidance, and feedback to help me grow throughout my professional journey. These individuals will help pave a stronger, more resilient path to success in my career. Finally, I would like to thank the staff at SCSU for providing valuable education to prepare me for where I am today.

Table of Contents

Chapter 1: Introduction

Chapter 1.1: Background

I am a graduate student at St. Cloud State University working towards obtaining a P.S.M. in Software Engineering. My academic career started with a bachelor's degree in electrical engineering at North Dakota State University. Following these studies, I desired a pivot to more software-oriented roles due to personal interest. This led to enrolling in the PSM Software engineering program with St. Cloud State University that I am now in the process of completing. The capstone requirement of this program has two variations that allow students to meet its requirements. The capstone choice that I have opted for is that of suitable work experience. This work experience was split out at two different workplaces across different semesters. These workplaces are TSI Inc. and MiTek. In addition to these two internship experiences with their respective tasks, a software project was developed with the purpose of highlighting skills and abilities developed because of the time spent at St. Cloud and the internships.

Chapter 1.2: About TSI

TSI Inc. is a globally recognized leader in the field of precision measurement instrumentation. Founded in 1961, the company has established itself as a trusted provider of solutions for a wide range of applications, including aerosol research, fluid dynamics, environmental monitoring, industrial hygiene, indoor air quality, and more. TSI was founded by Dr. Leroy Sheldon and Paul W. Studer in Minnesota under the name of "Thermo Systems Inc.". [1] The company initially focused on designing and manufacturing temperature sensors and instrumentation. Over the years, TSI's product portfolio expanded to encompass a broader spectrum of measurement technologies. TSI's key focus areas are as follows:

1. **Aerosol Research and Particle Instrumentation:** TSI is renowned for its expertise in aerosol research and particle instrumentation. The company's instruments are used to study the behavior and characteristics of particles in various environments, aiding in crucial research for health, environmental, and industrial applications.
2. **Fluid Dynamics and Flow Measurement:** TSI provides advanced flow measurement solutions for studying fluid dynamics. These solutions find applications in aerodynamics, automotive testing, HVAC (Heating, Ventilation, and Air Conditioning) systems, and other areas where understanding airflow is vital.
3. **Environmental Monitoring:** TSI's environmental monitoring solutions are instrumental in measuring air quality, emissions, and pollutants. The company's products are vital tools for government agencies, environmental researchers, and organizations committed to preserving the environment.
4. **Industrial Hygiene and Workplace Safety:** TSI manufactures instruments that ensure workplace safety by monitoring and measuring various industrial hygiene parameters. These include exposure to harmful substances, noise levels, and other occupational hazards.
5. Indoor Air Quality: TSI's offerings for indoor air quality measurement help identify and monitor pollutants within enclosed spaces, contributing to healthier indoor environments in residences, offices, schools, and public buildings.

TSI is committed to environmental sustainability and corporate responsibility. The company focuses on developing eco-friendly products and integrating sustainable practices in its operations. Additionally, TSI actively engages with local communities through various philanthropic initiatives, further reinforcing its dedication to making a positive impact beyond its

business endeavors. TSI stands as a pioneer and a trusted name in the field of precision measurement technology. With a legacy of innovation, a global presence, and a commitment to sustainability, TSI continues to shape industries and contribute to advancements in research and monitoring technologies.

During my tenure at TSI, I worked in a role directly supporting the manufacturing floor and was responsible for troubleshooting software/hardware issues globally, supporting legacy software, developing new software, and writing/performing software testing and validation plans. While I was exposed to many different software and hardware products, there are three very strong projects that would be better focused on as a highlight that took the bulk of my time and effort while at TSI. These projects include CFlow, Sensor Management Studio, and Sensor Database Annealing System. The projects will be elaborated upon further in the Project Overviews section of this document.

Chapter 1.3: About MiTek

MiTek Inc. is a global leader in providing advanced engineered products, software, and services for the building components industry. Established in 1955 and headquartered in Chesterfield, Missouri, MiTek has continuously pioneered innovative solutions that optimize the design, manufacture, and construction of buildings. MiTek's journey began with a focus on the development of metal-plate-connected wood trusses, a groundbreaking technology at the time. Over the years, the company expanded its product offerings to include software, engineered products, and systems that enhance efficiency and accuracy in the construction industry [2]. Today, MiTek operates in more than 40 countries and serves a wide range of industries, including residential and commercial construction, renovation, and industrial sectors. Some of MiTek’s product/services offered today include:

1. **Software Solutions:** MiTek is renowned for its innovative software solutions designed to streamline various processes within the construction industry. Their software offerings encompass design, engineering, estimating, project management, and collaboration tools. One of their flagship products is MiTek SAPPHIRE™ Structure, a comprehensive software for structural design and engineering. MiTek is currently in the process of improving BIM availability for homebuilders through software solutions.
2. **Engineered Building Products:** MiTek manufactures a diverse array of engineered building products, including connectors, fasteners, anchors, and structural components. Their products are designed to enhance the structural integrity and durability of buildings while optimizing construction efficiency.
3. **Building Systems:** MiTek provides prefabricated building systems that integrate seamlessly with their software, enabling efficient project planning and execution. These systems range from roof and floor trusses to wall panels, making construction faster and more precise.
4. **Capital Equipment:** The company offers automated equipment solutions that optimize the manufacturing of building components, making the production process more efficient and cost-effective.

MiTek collaborates with industry partners, academic institutions, and professionals to foster innovation and knowledge-sharing within the construction industry. These collaborations help drive advancements in technology and create a platform for the development of best practices. MiTek has revolutionized the construction industry through its commitment to innovation, technology, and sustainability. Their comprehensive suite of software, engineered products, and building systems empower builders, designers, and engineers to construct efficient, durable, and

environmentally responsible structures. As construction methods evolve, MiTek remains dedicated to leading the charge in shaping the future of construction through advanced solutions and industry partnerships. During my internship at MiTek I primarily contributed to one project called Array and will elaborate on it further in the next section. In addition to this project, I was responsible for managing Azure DevOps CI/CD pipelines for the team as well as analyzing software vulnerabilities reported within Veracode.

Chapter 2: Project Overviews

Chapter 2.1: CFlow

CFlow is an in-house software utilized by TSI to calibrate flowmeter's measurements as well as verify their accuracy. The software interfaces with a communication box to pass messages to a central communication system. This communication system provided access to precision voltage/current measurement systems, the flowmeter(s) under test, flowrate controller, pressure controller, and a digital valve actuator. The machine controlled allows for highly customized, highly complex calibration processes that varies from flowmeter model to flowmeter model. CFlow also stores all information surrounding a flowmeter's calibration and verification in a central database to allow easy quality audits, manufacturing engineer data dives, and software troubleshooting.

Technologies Used:

- Back-end: C++ , VISA-C (Keithley 3700A programming)
- Front-End: MFC (Microsoft Foundation Class)
- Data: Microsoft SQL Server, I2C, SPI, TCP/IP

Prior to the development of CFlow, TSI had no automated way to calibrate and verify each individual flowmeter by hand. The time required for each flowmeter on average neared two hours. Additionally, the floor for labor skill was greatly heightened due to the individual calibrating and verifying the devices needed to be able to control/measure all settings by hand, requiring a technician level 2 to be tied up during the calibration rather than a near-entry level position. Additionally, the manual nature of taking measurements and recording data was resulting in quality issues and gaps in missing data. While requiring a technician to manually calibrate an instrument is not uncommon for certain precision measurement tools that TSI offers,

TSI was also in talks of OEM (original equipment manufacturer) contracts with high profile companies. These talks would require TSI to produce more instruments than was currently possible. To that end, manufacturing engineers and software engineers identified a potential solution.

Manufacturing engineers drafted and built multiple calibration machines that would allow for automated control/measurement of all relevant physical variables and communicate with TSI's flowmeters using SPI, I2C, TCP/IP to communicate with the flowmeters with the communication protocol switching based on the device, with the idea that CFlow would control and monitor the calibration and verification process. The result of CFlow was that the average calibration time was cut down to 22 minutes while also allowing up to 6 devices to be calibrated at once, depending on the model. Additionally, the skill floor of labor was now lowered to that of someone going through high school as all that was required was connecting these devices to the calibration machines and scanning the device barcode. These laborers would also be able to perform other tasks while the flowmeter calibrations ran, saving huge costs on labor while still improving efficiency. The data collected during each run is instrumental in allowing deep dives, and now that CFlow was collecting all information for the calibration, there was no room for missing or false data while also being easily accessible by anyone with T-SQL proficiency. CFlow would go on to push TSI's flowmeter department to gross multi hundred million dollars a year.

Chapter 2.2: Sensor Database Annealing System

Sensor Database Annealing System (SDAS) is a desktop-based software used to measure sensor and thermistor voltages. Sensors must undergo a 14-day annealing process in order to be properly prepared to be inserted into flowmeters. If the sensors or thermistors fail a select

number of checks, the sensor must be discarded to prevent faulty products or wasted calibration time. The sensor must pass through resistance checks 0 degrees C, 80 degrees C, and room temperature to be considered. All data collected is stored in an SQL Server database where the software can automate and produce reports on sensor status and statistics.

Technologies Used:

- Back-end: C++, VISA-C (Keithley 3700A programming)
- Front-End: QT [3]
- Data: SQL Server

SDAS was one of the first software built at TSI for the sole purpose of calibration/verification purposes. The sensors and thermistors that the software validates/controls are used throughout the manufacturing floor by different products. Without these sensors, many products cannot be produced and lead to production blockages. However, the software was left in a long-term maintenance state where no developers had touched it since it's development in the 90s. This came to a head when the software would no longer function on newer operating systems and a dwindling supply of suitable replacement computers due to the general age of systems required to run the software. As a result, I was tasked with rebuilding the legacy software using modern frameworks.

Aside from the general requirement of being overhauled to work on modern windows operating systems, management had identified areas of improvement to increase sensor throughput while allowing flexibility of the system in the future. Part of this included exchanging the sensor measurement device with a more modern counterpart that had far more measurement channels. This increase in measurement channels allowed us to ramp up and down on the number of sensors being measured at any given time. An additional modification was to modify how

sensor types were handled as the base software was very rigid and needed software updated to handle new sensor types. By having a general sensor model and pulling the sensor's information from the database, we were able to create a system that both new product development and production were able to take full advantage of. These changes, paired with the increased throughput, led to a far more user friendly and quality-controlled experience while having the feel of a modern desktop application.

The software was a large success with its end users with reports of faster operation and general morale improvements due to positive change. Though, it is unclear whether or not that was simply due to the newer and faster computers that were installed shortly after. What is clear, however, was the output benefits. The individual number of sensors that could be ran per test went from 16 to 24, a 50% increase. Additionally, the software was modified to allow operators to run two sets of sensors at a time in the different baths. This effectively means that the process went from 16 to 48 sensors, a 300% output increase. This blew the lid off of a process that for a long time had been a bottleneck for production. While the system did not need to be running 24/7 anymore, it no longer needed to, which in turn allowed enhanced flexibility and faster turnaround times on sensor requests.

Chapter 2.3: Calibration Time Analyzer

Calibration Time Analyzer (CTA) is a web-based application developed using ASP.NET Core and Razor Pages. Its primary function is to monitor and record the uptime and downtime of calibrators based on data stored in a database. This software offers a user-friendly interface for efficient tracking and analysis of calibrator performance, aiding in maintenance and decision-making processes. Additionally, this was used in identifying and correcting any unnecessary downtime in the calibrators.

Technologies Used:

- Back-end: C#, ASP.NET Core [4],
- Front-End: Razor Pages, HTML, CSS, JavaScript
- Data: SQL Server, Entity Framework Core [5]

When COVID-19 ramped up, TSI saw a ramp up in particle measurement and flowmeter devices due to their close relationship with the medical field. The increased demand caused TSI to bring on a temporary 3rd shift as well as a weekend shift just to try to keep up. However, production managers noticed that they were consistently running behind expectations and weren't entirely sure why. While the calibration performance data already existed and could be extrapolated from the database it was hard to visualize, the calibration results were not intuitive to interpret, and there were many edge cases to consider. To that end, it was determined that a software project to ease these data-delving pains and provide this data in a simple format to any user who might want it was necessary.

CTA seamlessly integrates with a database to retrieve and store information related to calibrator uptime, downtime, and performance metrics through SQL Server and Entity Framework. The software presents a comprehensive dashboard with intuitive graphical representations, offering an at-a-glance view of calibrator statuses, uptime percentages, and historical performance trends via user desired graphical formats such as pie charts or bar charts in hour, daily, and weekly time lengths. Users could easily add, edit, and delete calibrator records, inputting relevant details such as model, calibration bench, location, and other essential information. Additionally, the software was built with scalability in mind to handle a growing number of calibrators and users, ensuring consistent performance as the database and usage scale over time. Any given user was also able search and filter calibrators based on various

parameters, facilitating easy navigation and access to specific calibrator records. The software would generate detailed reports and analytical insights based on historical uptime and downtime data, enabling data-driven decision-making for calibrator maintenance and optimization.

The software successfully allowed production managers and manufacturing engineers to identify general downtime due to laborer inefficiencies as well as common points of failure and introduced procedure updates to fix these issues. On average, nearly 135 minutes per calibrator per day would wind up being reclaimed, approximately a 9% output increase. This optimization, paired with other improvements at the time, allowed TSI to meet their quota and fulfil their contracts. After the COVID-19 pandemic slowed down and TSI no longer needed to be near 100% capacity, product managers still opted to use the software due to the value of the information it provided with the minimal effort required to parse it.

Chapter 2.4: Array

Array is an add-in product for Autodesk Revit developed by MiTek aimed to simplify BIM for homebuilders. Its core functionality is the ability to easily achieve a lot specific solve, which is the creation of a customized building model derived from a base model with various options selected. The software add-in aims to allow experienced architects and Revit users to have their existing models converted into a BIM format without manual intervention and aims to have the users complete their first solve within thirty minutes of using the add-in. Currently, MiTek Array has been in development since April of 2022. The target release date for Array is November 1st, 2023.

Technologies Used:

- Back-end: C#, .NET Framework 4.8, Revit API, JavaScript, Node JS, NPM, Swagger
- Front-End: WPF, DevExpress

- Data: SQL Server, Entity Framework

MiTek has been no stranger to pushing the construction industry forward through innovation and optimization since its founding in 1955. With each development in the construction industry, MiTek has been quick to step in and capitalize on new opportunities to drive these innovations. With the rising popularity of, and Autodesk's acquisition of the product, Revit was no exception with an estimated 73% of users using the product [6]. MiTek had conducted a study and found that while Revit was predominantly designed as a BIM software, homebuilders rarely used it for BIM purposes due to the technical and time tradeoff of such not being worth it for the smaller buildings that homes normally were. Additionally, since MiTek had a similar add-in product for AutoDesk Architecture with their Blackpoint software, MiTek found themselves with the necessary subject matter expertise to capitalize on the opportunity.

Array has two core components with specific functionality:

- A web API to handle licenses and data delivery while facilitating multiple users from remote locations.
- The desktop add-in to receive data and perform the work of transforming the BIM process for users within Revit.

The web API uses the entity framework to perform database transactions using modelled objects on a database set up at run time. This modelling has been made trivial due to the use of Entity Framework and its migration tools allowing us to make swift, code-first changes. Swagger provides a simplistic and easy to use web interface to handle calls to the web API.

The desktop side of the Array is used to interface with Revit and perform the work of being the add-in. Users will begin by setting up their base Revit home model. Following this, Revit users would normally use design options to create building variations that lack depth while

also using the tedious built-in BIM components within Revit that will not update as the user makes changes. This is where Array steps in for the user to improve their efficiency and options. Using Array, the user can create multi-layered, building component that can seamlessly be switched in and out with other matching components. These components can have custom logic assigned to by the user that will be evaluated upon attempting to make a customized, lot solve of the base model – the result of which is a tailored building model based on the users' desires. Additionally, using the same custom logic functionality, users can also modify the properties of any selected element within Revit to suit their needs. The result of which is a highly complex, highly automated system that does as much work as possible for the user.

As of writing, MiTek has 4 out of the ten largest homebuilders in the United States as alpha partners. These homebuilders are eager for the improvements made in our 1.0 release and have already pledged to purchase the licenses on release. Additionally, internally, MiTek upper management has greenlit the continued development of Array due to the perceived value of the product. The team is also looking to grow to meet customer demand and support criteria. MiTek aims to have 40 of the top 100 home builders using the software by November 2024.

Chapter 3: Course work

1. **SE550 - Software Reverse Engineering:** SE550 is a course that trained me to analyze and understand software that I may not have access to the source code for. An underlying takeaway from this course was memory management and how to protect classified information from individuals looking to exploit credentials stored in code. The knowledge gained from this course was helpful during my internships as it allowed me to write, or upgrade existing, code that was safer from potential cyber attackers.
2. **SE641 - Application and Database Systems:** During my internship, there were many points where I had to interact with database systems – be that design, implementation, modifications, and so on. This course gave great insight into database systems by allowing me to perform my tasks during internship in efficient, well-informed ways.
3. **SE565 - Software Design:** Software architecture is a crucial part of managing a successful project. After finishing this course, I felt prepared and ready to deliver a presentation on common software design patterns to the rest of our team. These design patterns, such as singletons, observers, factories, and so on, were then implemented to improve code use and reusability throughout projects during the internship.

Chapter 4: Revit Unit Tester

Due to non-disclosure agreements and proprietary software property, I was unable to turn in any source code or design documents. During the preliminary proposal meeting, the request deliverables were a Github link to the a full project/source code and a step-by-step guide on how to use the software. As mentioned prior, the project has an emphasis on the skills learned from my internship during MiTek as I believe this experience was most relevant. To that end, the project I have implemented is a reuseable, general purpose Revit Unit Testing Tool.

Chapter 4.1: Revit Unit Tester

The Revit Unit Tester was designed after noticing that our team had no way to automate and test code that interacted with the Revit API, particularly the events driven by the user when opening, modifying, and saving Revit projects. As a result, it can be difficult, or impossible, to write unit tests while developing Revit Addin software to create full code test coverage. Normally, this would result in the need to manually run these test cases, causing a bloated need for software quality resources. This is where the need and value of the Revit Unit Tester comes in as it provides a way for developers to write functional and unit tests in separate DLL packages, provide the DLLs and any required Revit Projects to the Revit Unit Tester, and the unit tester will allow the user to run any tests present within the DLL as the user selects.

For a more detailed explanation on how to use this software, please refer to this video: https://youtu.be/Pj4TqwaoP6k

In addition to this, the code can be found located in a github repository here: https://github.com/Gimmacus/SCSUThesisProject

Technologies Used:

- Back-end: C#, .NET Framework 4.8, Revit API
- Front-End: WPF, DevExpress

Chapter 4.2: Deliverables Remarks

With this supplementary project, I believe I have demonstrated progression of skills and abilities while solidifying my internship experience as a capstone to my PSM degree. While this software does not utilize many software frameworks, it makes up for it in technical depth paired with a legitimate need real world need. This tool can reduce overhead for Revit Addin software projects by reducing required testing resources as well as simple bulk testing setup. The team I interned with was greatly interested in the value of the tool and has begun the process of introducing it into their workflow to reduce testing efforts. This has allowed multiple testers on the team to refocus their efforts to other tasks, while reducing regression testing feedback from multiple weeks to a maximum of two days. The juxtaposition of upfront developer time to write the test cases and the resource savings for testing has made the tool a resounding success.

Chapter 5: Closing Thoughts

Software internships offer a multitude of benefits that can significantly impact a student or entry-level professional's career. With every project I put effort into throughout my internships, I grew my skills and made valuable connections with my peers. While not every hour spent was put towards a cutting-edge research project, there were still many takeaways that I believe an internship capstone provides over a project capstone. Here are some of the advantages of pursuing a software internship that I noted:

1. **Hands-on Experience:** Software internships provide valuable hands-on experience, allowing interns to apply theoretical knowledge to real-world projects. This practical exposure enhances their understanding and competence in various programming languages, frameworks, and tools.
2. **Skill Development:** Internships foster skill development across a spectrum of areas such as coding, debugging, teamwork, project management, and communication. Interns often get the chance to work with cutting-edge technologies, enhancing their skillset and making them more marketable to potential employers.
3. **Networking Opportunities:** Internships offer a unique platform to build a professional network within the industry. Interns interact with seasoned professionals, mentors, and fellow interns, creating meaningful connections that may lead to job opportunities, mentorship, or collaborations in the future.
4. **Resume Enhancement:** Having a software internship on your resume demonstrates practical experience and dedication to prospective employers. It validates your commitment to the field and gives you a competitive edge in a crowded job market.
5. **Career Path Clarification:** Internships provide insight into different roles and career paths within the software industry. Interns can explore various domains, allowing them to

make informed decisions about their future career direction based on their interests and strengths.

6. **Company Culture Understanding:** Internships offer an inside look into a company's culture, work environment, and expectations. This exposure helps interns determine if they resonate with the culture and values of the organization, aiding them in making informed decisions about future employment.
7. **Potential Job Offers:** Many companies use internships as a way to evaluate potential future employees. A successful internship often translates into a job offer, providing a direct pathway to full-time employment after graduation.
8. **Learning from Professionals:** Interns have the opportunity to learn from experienced professionals in the industry, gaining insights into best practices, industry trends, and real-world problem-solving approaches. This mentorship is invaluable in shaping a successful career.

In conclusion, software internships play a vital role in shaping the careers of aspiring software developers. They offer practical experience, skill development, networking opportunities, resume enhancement, career clarity, cultural understanding, and the potential for future job offers. These advantages make software internships a crucial steppingstone towards a successful and fulfilling career in the software industry.

References

[1] “TSI’s history,” TSI, https://tsi.com/discover-tsi/tsi-s-history/ (accessed Sep. 21, 2023).

[2] “About MiTek,” MiTek Inc., https://www.mii.com/about/ (accessed Sep. 24, 2023).

[3] “Qt | Development Framework for Cross-platform Applications,” www.qt.io. https://www.qt.io/product/framework?utm_term=c%2B%2B%20development%20software&utm_campaign=Development (accessed Sep. 27, 2023).

[4] T. Dykstra, “Overview of ASP.NET Core,” learn.microsoft.com, Sep. 22, 2023. https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0 (accessed Sep. 27, 2023).

[5] A. Vickers, “Overview of Entity Framework Core - EF Core,” learn.microsoft.com, May 25, 2021. https://learn.microsoft.com/en-us/ef/core/ (accessed Sep. 27, 2023).

[6] “Benefits of REVIT Modeling Services for Designing,” *Indovance Inc*, Dec. 05, 2022. https://www.indovance.com/knowledge-center/benefits-of-revit-modeling-services-for-designing/#:~:text=Revit%2C%20one%20of%20the%20most