Paper ID #20374

# Student Paper: Small Team Agile Systems Engineering For Rapid Prototyping of Robotic Systems

**Mr. Charles Avery Noren, Texas A&M University Vehicle Systems & Control Laboratory**

Charles Noren is an undergraduate research assistant at the Texas A&M University Vehicle Systems & Control Laboratory and task leader for the rail-based robotic system project. He is expected to graduate with a Bachelor of Science in Aerospace Engineering in May of 2018, and plans to continue his education at Texas A&M University with a Master of Science in Aerospace Engineering.

**Kendra Lynne Andersen, Texas A&M University**

Kendra Andersen is an undergraduate student in electrical engineering, graduating from Texas A&M University in May 2018. She plans to work in the aerospace industry after graduation, and takes language courses in order to accomplish her ambition of becoming a polyglot.

**Ms. Kanika Gakhar, Texas A&M University Vehicle Systems and Control Laboratory**

Kanika Gakhar is an Undergraduate Research Assistant at the Advanced Vertical Flight Lab at Texas A&M University. She is currently pursuing a degree in Aerospace Engineering and plans to graduate in May 2018.

**Angela Olinger, Texas A&M University**

Angela Olinger is an undergraduate mechanical engineering student at Texas A&M University graduating in May 2018. She plans on completing her Master's of Science degree after graduation before going on to work with materials design in the space industry. She is also a classically trained ballet dancer and continues to enjoy dancing and choreographing.

**Mr. Preetam Palchuru, Texas A&M University Vehicle Systems & Control Laboratory**

Preetam Palchuru is an Undergraduate student studying at Texas A&M University. He will graduate in May 2018 with a B.S. in Aerospace Engineering. He currently works at the Vehicle Systems and Control Laboratory as an undergraduate researcher. After finishing his undergraduate education, he plans on getting a Master of Science in Aerospace Engineering.

**Mr. Scott Thien Tran, Texas A&M University Vehicle Systems & Control Laboratory**

Scott Tran is an undergraduate research assistant at the Texas A&M University Vehicle Systems & Control Laboratory. Expected to graduate with a Bachelor of Science in May 2018.

**Dr. John Valasek, Texas A&M University**

John Valasek is the Thaman Professor of Undergraduate Teaching Excellence, and Director, Center for Autonomous Vehicles and Sensor Systems (CANVASS), Director, Vehicle Systems & Control Laboratory, Professor of Aerospace Engineering, and member of the Honors Faculty at Texas A&M University. He teaches courses in Aircraft Design, Atmospheric Flight Mechanics, Modern Control of Aerospace Systems, Vehicle Management Systems, and Cockpit Systems & Displays. John created the senior/graduate level course AERO 445 Vehicle Management Systems, which was the first regularly offered course on this topic in a U.S. aerospace engineering department.

John earned the B.S. degree in Aerospace Engineering from California State Polytechnic University, Pomona in 1986 and the M.S. degree with honors and the Ph.D. in Aerospace Engineering from the University of Kansas, in 1990 and 1995 respectively.

# Small Team Agile Systems Engineering For Rapid Prototyping of Robotic Systems

Engineering management decisions play critical roles in the ultimate success or failure of a project. The approach taken, the team size, the deliverables and their respective due dates, and the environment are key factors which engineering managers, technical team leads, and educators must all address to complete an assigned goal. While many courses at both the high school and college level focus on achieving set educational outcomes, the manner by which these outcomes are achieved may prove difficult to select due to the dynamic nature of the classroom. From the observations of this team, high school and university project deliverables and educational expectations are commonly, and sufficiently, satisfied by a small team of individuals, and thus it is the focus of this paper to discuss the group dynamics and structure, and the advantages and disadvantages, of the small team approach to engineering challenges. In addition to addressing the role of team size on project completion, an investigation into benefits of the agile systems engineering approach towards technical projects will be discussed with specific regard to the deliverables and time management, learning outcomes, long term project success, knowledge retention, and the learning environment. Finally, it is the goal of this paper to also highlight from the perspective of the students the benefits and challenges of working on a small team with an agile systems approach towards difficult technical challenges.

## Introduction

Early in the summer of 2016, an engineering team was formed to address the challenge of developing an autonomous rail vehicle (Railbot). Initial work focused on the Concept of Operations (CONOPS) and Requirements of a full scale vehicle and then progressed to focus on a small demonstration vehicle. During the formulation of the vehicle requirements and creation of the demonstration vehicle, the program manager and principal investigator discussed the challenges of such a program. It was anticipated that such a program would require the skills of many different engineering fields, including Aerospace, Electrical, and Mechanical. As such, this team was formulated under the guidelines of the AggiE-Challenge Program, a specialized environment developed at Texas A&M University (TAMU) to address multi-disciplinary projects. The AggiE-Challenge program served as the basis from which this team was developed, and as such the team follows many guidelines that are listed in Lagoudas and Froyd's [1] work on multidisciplinary teams. Some of these guidelines include: Small Team size, Multi-disciplinary Team Construction, and Faculty, Industry, and Graduate Student team support. In short, this team may be seen as an instance of the AggiE-Challenge program as the benefits derived reflect those discussed in Lagoudas and Froyd's work [1]. In addition to this framework, a semester long research course focusing on aspects of Systems Engineering (SE) similar to that taken in Valasek and Shyrock's work [2] on capstone design at TAMU was adopted for the Railbot program.

**Team Size: Small Team**

It has been the experience of the authors that many engineering courses have difficulty maintaining an optimal balance between the overall benefits gained by students participating on the team and the number of students working on the project. While the scale of the challenge provides guidance to many programs, it has been found that even a slight mismatch in workload may cause the program to fail to engage all individuals on the team. For larger engineering classes, such as the introductory level engineering classes at TAMU, the multidisciplinary aspect of the proposed challenges are lost on the fact that the engineering challenge is not large enough to support the participation of multiple individuals. In addition, the multidisciplinary teaming narrative fails to materialize as the individuals who specialize in different aspects of the project dominate that aspect without requesting guidance or input from other team members.

For this program, a small team size of seven members ensured engagement of all team members and required different engineering disciplines to work together to develop a working system. In order to best distribute workload, this small team was further divided into working groups. These working groups consisted of two engineers from different majors and were assigned different tasks based on three categories: Vehicle, Electronics and Sensors, and Simulation and Control. These separate teams worked in parallel to develop the necessary project deliverables.

| Railbot | | | | |
|---|---|---|---|---|
| | Team A Objectives | Team B Objectives | Team C Objectives | Everyone |
| Updated 10/26/2016 | Chasis and Integration | Sensors, Electronics, and Software | Simulation, Software, and Control | |
| Date | KEY: | Completed | In Progress | Planned |
| October 12 | N/A | N/A | N/A | Complete the concept of operations, defined requirements, derived requirements, and inital investigation of subsystem components |
| October 19 | Define Project Deliverables | Investigate and recommend if railbot should use TI chips | N/A | |
| | Define Demonstration Critical Success Measures | Investigate and recommend 2 methods of determining relative positioning | | |
| | Get customer input of the above | ~~Investigate and recommend other sensors needed for fulfilling the derived requirements~~ | | |
| October 26 | Select 3 chasis and the necessary parts that will be needed to modify it | Select processor family and development enviornment. Pass details along to team A. | Model system dynamics of train-railbot systems in MATLAB. For starters we probably want to see if there is anything in the literature about simple rail car modelling otherwise we will think of something ourselves. | Investigate and recommend communications links (full scale and demo) and send them to the customer |
| | Begin design of system architecture based off of high level subsytem requirements (form factor, power draw, etc) Use the requirement to guide this! | Send selected processor to the customer for feedback | Investigate and recommend other sensors needed for fulfilling the derived requirements (This is basically a double check). | |
| | | Look into details of sponsorship with the company that sells the $4K INS | | |
| | | Recommend best RFID reader and tags for our application. Pass details along to team A. | | |
| November 2 | Design power solution for the demonstration system | Define initial hardware, communication, and software architectures | Model system dynamics of demo systems | |

Figure 1: Early Example of the Working Group Task Chart

Advantages of this method included decreased subsection development time and increased information flow. Each working group served as a node in the group communication network. Instead of relying on individuals, the communication nodes were responsible for task implementation and completion. As two people composed one team, accountability transferred to the team rather than to any individual. In addition, any one individual was not integral to the critical path, as both engineers in the working group were responsible for the knowledge of the group. By developing these working groups, action items along the critical path, such as those listed in Figure 1, could be addressed in a timely manner, and any specific individual's schedule did not impede the critical path.

One challenge of the working group approach is that an "Integrations" team needs to be assigned early. Integrating each of the subteam pieces proved to be one of the most time consuming and challenging portions of this engineering challenge. In addition, individuals on the integrations team must have a deep understanding that constantly updates as each working group iterates through their design. Tools such as GitHub and Google Drive assisted with this aspect of the engineering design as version control and up to date documentation was critical to system functionality.

**Approach Taken: Systems Engineering**

Systems engineering, in practice, is the succession of several structured design phases planned with a goal of designing a complete system through the fulfillment of system requirements [4]. In addition to the creation of a product or system, the action of following the systems engineering process encourages the development of formalized documentation in order to reflect on design success. This documentation includes meeting notes, requirement tables, time tables, and team engineering decisions. In essence, systems engineering is a top-down approach to completing a project, starting from the most abstract idea and forming more concrete definitions until all that remains is the actual implementation and testing of the designed system.

The process begins with a general concept, generally marked by creation of a Concept of Operations, then moves into the conceptual design, which is a more detailed listing of the general goals and how they might be accomplished. The third stage is the preliminary design, in which the concept is transformed into a full list of requirements and subsystems are arranged to implement those requirements.An example of a list of requirements and test metrics may be seen in Figure 2. Test metrics are made for each of the requirements to ensure that the system performs to its specifications. The last stage is the detailed design, where the requirements are fully implemented in production for testing purposes. This is when the testing plans are fleshed out and the subsystems implemented.

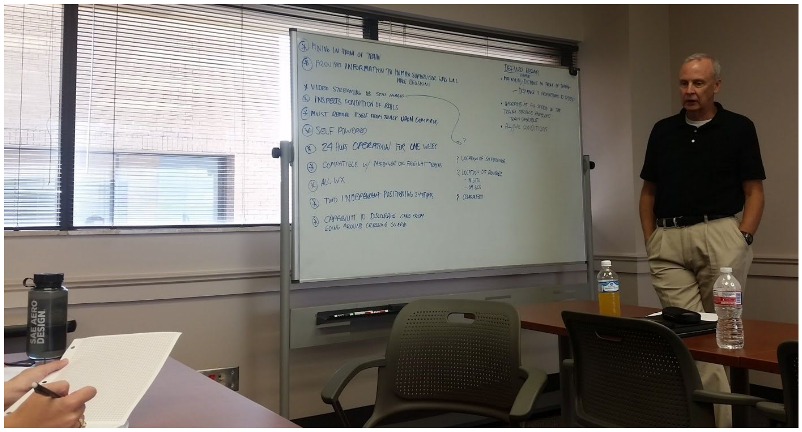

Figure 2 - A discussion about system requirements

The systems engineering strategy can be applied to many different sizes and types of projects, from company-wide development of large and complex systems, to small team projects in a classroom environment.

Agile systems engineering is a subset of systems engineering in which a specific style of project management has been applied. In general, an agile approach means that the design development is segmented into many small tasks, and the process of the entire system is frequently reevaluated in general team meetings. This approach encourages personal interactions and frequent communication between the customer and the developers, as described by Johnson's presentation [3]. In addition, agile systems engineering precipitates extensive planning before accomplishing the set tasks expeditiously and efficiently.

The approach taken by the team developing Railbot closely follows the systems engineering process with a focus on agile management. In the first stage of the process, team members worked closely with the customer who instigated the AggiE-Challenge to develop a Concept of Operations which concisely stated the needs of the autonomous vehicle. The next steps were to develop an account of general requirements while beginning research into what types of tools and available subsystems would be satisfactory, maintaining regular but less frequent communication with the customer. The last phase before actual implementation of the system comprised of establishing specific and measurable test metrics to achieve the detailed requirements which had been generated. Throughout this process, the team members, program manager, and customer spoke routinely in order to ensure that the project being developed remained reasonable in scope and satisfied the desires of the customer. This example demonstrates fluidly how agile management and systems engineering can mesh together to form an effective project development strategy.

When the implementation portion began, key facets of agile systems engineering began to take their most obvious effect. The team met regularly to assess what had been done in the last week and set objectives for the week to come, allowing working groups of two or three members to complete their objective when convenient before the next team gathering. These working groups changed frequently, as different segments of the project required varying skills to complete. This approach allowed all of the work to successfully contribute towards the goal of creating an autonomously operating rail vehicle, while also enabling parallel timelines and allowing every team member's skill sets to be used in the most appropriate manner. The final stages of the project involved preparations for a final demonstration which reflects an adherence to the test criteria. Since these stages mostly consisted of the integration of subsystems that had been developed during earlier stages in parallel, the team met met as a whole unit during the last few stages. At this point, working groups still focused on specific integration tasks, but also facilitated communication where necessary. The team members continued to meet periodically. However, instead of meeting weekly, the team increased its frequency of meetings and met every couple of hours. This allowed rapid development of the final product in a much shorter timeline than would have been possible had the planning stages of agile systems engineering not been performed.

**Multidisciplinary Learning**

Many engineering projects on university campuses are discipline or major specific. This forces students to only interact with other students who have similar backgrounds and ways of thinking. However, by incorporating a variety of engineering majors such as mechanical, electrical, and aerospace, the team established a collaborative multi-disciplinary learning environment. By synergizing ideas and efforts, the team was able to produce results that were much better than those that could be produced by individual major-specific groups.

The team members' different coursework and wide variety of project experience allowed the team to adopt a diverse set of approaches in order to solve the same problem. For instance, two different methods were considered in order to ensure that the railbot travels in a straight line: a software route as well as a mechanical route. Although the software route was ultimately implemented, it helped to have an alternative solution and the ability to collaborate with engineers of different disciplines in order to narrow down the choices to the most optimal solution.

In addition to belonging to different majors, the members of the team have a diverse set of interests and skill-sets within their respective majors. As a result, the team members had opportunities to share their expertise and learn from each other. While some members were heavily interested in controls, simulation, and software, others were more inclined towards structural design and aerodynamics. Nonetheless, by the end of this project, all the team members obtained a thorough understanding of the various mechanical and as well as information communication aspects of autonomous vehicle control and dynamics. Additionally, this project gave team members an opportunity to indulge in disciplines that they had always been intrigued by, but never had the chance to formally pursue. For instance, one of the team members had always been fascinated by animation and graphic design; this project gave her a chance to finally learn how to convert mathematical simulations into visually appealing animations. Therefore, this multidisciplinary project is giving students a chance to step outside their comfort zone, learn from each other, and develop synergistic approaches to problem solving.

**Deliverables**

Most engineering-oriented projects tend to be broken up into pieces. Each piece builds on each other and eventually leads to the final product. At each stage, the team has deliverables to show before the next step can take place. In a normal systems engineering approach, there are a small number of key deliverables for a few groups at each step. With agile systems engineering, the number of deliverables increases significantly. Agile systems engineering requires a lot more thought and planning in the initial stages. In this project, deliverables in the initial stages included timelines (Gantt charts, etc.), thorough investigations of systems, and requirements. In fact, a lot of attention and time was given to defined and derived requirements for the project. Each week, members were responsible for researching and outlining the requirements in order for the end goal to be deemed "successful." Deliverables were more short term (weekly or even within a few days) and plentiful than with a normal systems engineering approach which involves fewer deliverables that are long-term oriented.

Once the requirements and planning stage was complete, the team had to begin building the system. The deliverables created during this period were starkly different from the initial phase of the project. They were more concrete in nature. Certain specific tasks needed to get done. Tasks like programming, hardware testing, and algorithm creation were broken up into small chunks and assigned to individuals or working groups. As seen in Figure 3, the system considered in this engineering challenge included microcontrollers, structural elements, and power distribution. The deliverables of the project had weekly or even daily deadlines.

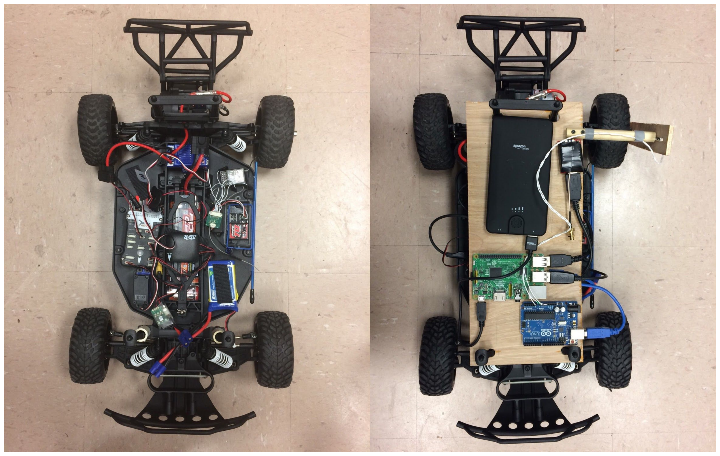

Figure 3 - System Layout depicting Mechanical and Electrical Challenges

Agile systems engineering also requires much more communication and accountability than normal systems engineering. The team used tools such as cloud storage, Slack and Github to keep track of progress and assign new deliverables to members. Since there are always more moving parts in agile systems engineering, it is vital to have continual communication about deliverables between team members. One neat feature of agile systems engineering is how flexible deadlines can be. Since each deadline is so short and the deliverables are fairly small each time, the deadlines are relatively flexible (a day or two), allowing for troubleshooting and quality assurance.

Finally, when necessary, it is also possible to momentarily break up working groups for a phase of the design. One example of this was observed during the hardware testing phase, where one of the structural and mechanical systems engineers assisted the sensors and electronics subteam testing by piloting the Railbot. As seen in Figure 4, a group of individuals is need for ground testing, and thus being able to strategically recombine the working groups to optimal testing teams assisted in the completion of the project.

Figure 4 - Vehicle Testing

**Environment**

The small size of the team allowed the project to exist in a unique working environment, specifically affording flexibility in both scheduling and physical working location. Every member of the team had other responsibilities to balance with the project, which made scheduling a challenge. However, this required each team member to be mindful of completing their tasks on time so as not to overburden any other team member's schedule.

In the early stages of the project, the team worked in tandem developing requirements and preliminary design ideas. This phase of the project took place in a more traditional classroom setting. As the project moved on into the design and implementation phases, it became necessary for the team to be divided into smaller working groups. During this phase, the whole team met weekly to discuss the progress made and future steps. However, most of the actual design work was done by these small groups outside of regular meetings. This allowed subteams to work for longer periods of time while coordinating fewer schedules, ultimately allowing for more productivity. The model also afforded each subteam the opportunity to work in the physical environment best suited for a particular task. For example, software development typically took place in a more traditional computer laboratory setting at the Vehicle Systems and Control Laboratory, while fabrication and hardware integration took place in the Engineering Innovation Center (EIC) on the TAMU campus. The EIC is a facility that gives undergraduate students access to a machine shop, tools, and electronics equipment. This resource enabled the team to more quickly prototype hardware and test software without necessarily waiting for electronics to be ordered.

The subsystem model allowed scheduling to vary with the needs of the team and coursework demands of individual team members, but ultimately led to delays in interfacing between subsystems since individuals working on different tasks were not immediately available to answer questions. During the latter stage of the project, the team transitioned to a more collaborative working environment. While still maintaining the subteam model, the entire team met for longer periods of time, in the same location. This allowed the team to meet briefly to establish immediate goals, then work semi-independently, while conducting brief meetings every few hours to keep everyone on task and handle any unforeseen problems. This model ultimately allowed the team to work much more efficiently, especially because of the interdependencies of each subsystem. However, this model, as executed, was deemed to be impractical in the long run and was only considered possible during a period of intense work, immediately after classes had ended for the semester.

**Conclusions**

The decision to use an agile systems engineering approach through a small team has allowed the authors to take an abstract idea and create a prototype within a semester. Benefits and challenges of a small engineering team using the agile systems engineering method include:

1. The small team organization allowed a unique team culture where team members from different disciplines are able teach and learn from each other. Due to the nature of the challenge, a multidisciplinary team allowed team members to come up with diverse solutions to tasks because of each member's past project experiences and coursework.
2. The team's decision to divide into subteams allowed the project to progress despite conflicting schedules. Even if one member was unable to attend a meeting to summarize the tasks completed at least one other member had the same knowledge to keep the project flowing smoothly.
3. Regardless of the short term success of the challenge, long term continuing work using this model can lead to inefficiencies as current team members may be unable to predict future problems. If there is an inability for current engineers to meet up with future workers on the project, the system engineering documentation will serve as the only link between the programs. This could lead to possible misunderstandings and communication gaps. The system engineering approach suggests a legacy of continued development. As such, each engineering decision must be documented extensively to express the logic of the choice.

The student responses on the following page reflect the opinions of each engineer who participated in the program.

**Student Responses:**

**-Team Leader, Junior, Aerospace Engineering Major**
"The System Engineering approach to developing an engineering solution provides an overall context and framework to the engineering design process. As a student who has participated on large engineering teams and small engineering teams, I find that a systems engineering approach assists in considering the criticality of time and limited resources and administrative constraints generally lost in the process of design. Building within a team's limitations is a key factor in developing a system which meets the sponsor's criteria, but also meets a time table. Implementing the systems engineering approach is fairly simple, but it is extremely easy to become caught up in the details of the engineering requirements. At some point, the engineering team leader, or the instructor advising the team, must recognize that continued requirement edits cut into development time and can cause a lag in the deliverables. As such, there is a fine balance between spending too little or too much on requirements, and that line was not clear in the program. In addition, it is difficult to keep the entire engineering team engaged in the project development if too much time is spent on developing requirements. Many times, keeping the team motivated proves to be the deciding factor between project completion and project failure. A careful line must be walked to ensure that the team remains inspired while appropriating the correct amount of time to completing the engineering project."

**-Team Member, Junior, Aerospace Engineering Major**
"Much of the success of a systems engineering-focused project relies on predicting the unpredictable roadblocks for a project. During the design phase, the team needs to pay careful attention to specific time sinks and the projected timeline. Within this specific project, the team realized very far along into the project that the programming component of the project was significantly more time consuming than the hardware portion of the project that the team almost entirely focused on at the beginning of the project. If the team could do this again, we would spend significantly less time on designing the hardware of the system (while retaining robustness of design) and devote more time to developing the software package to control and test the hardware and the developed controller. In my opinion, the ability to predict these pitfalls is crucial to the success of a systems engineering project. Having people from multiple disciplines boosts the ability to foresee certain problems and their respective solutions."

**-Team Member, Junior, Electrical Engineering Major**
"The small size of the team allowed quick resolutions when roadblocks occurred as the project continued. When roadblocks arise the multidisciplinary knowledge of the team allowed team members to refer to each other on topics unfamiliar or not taught in their curriculum. Although some solutions were only quick fixes and present other roadblocks in later parts of the project from oversight when one person took charge."

**-Team Member, Junior, Aerospace Engineering Major**
"Right from the beginning, every deliverable for the project was presented along with a checklist that indicated which design requirements were being fulfilled by the respective deliverable. This approach helped the team use industrial standards for systems engineering and ensure that every minute spent working on the project contributed directly towards one of the final goals set by the customer-engineer interaction. Moreover, it served as an effective communication technique so that every team member knew exactly where he or she, as well as the rest of the team members, stood with respect to the overall project."
"Having had a chance to work on animations gave me a chance to step outside the boundaries of my major and experiment with new aspects of project management. I had a chance to climb outside of the engineering bubble and assume a creative role while considering the visual aspects that help convey the technological prowess of our product. Moreover, I learned to consider the customer's perspective and the factors that the customer would find most appealing. As a result, I had to force myself to see the product from the customer's eyes. This helped me manipulate the animation such that the customer would see all the salient features stand out in a visually appealing fashion, complementing the technological advantage of the product."

**-Team Member, Junior, Electrical Engineering Major**
"Overall, the small multidisciplinary team selected to complete the project proved to be appropriate in size and range of skills. While more foresight could have been useful in constructing the original project timeline, having multiple people capable of doing any portion of the project assisted in completing tasks quickly."

**-Team Member, Junior, Mechanical Engineering Major**
"While overall the systems engineering approach was beneficial to the project, the testing and integration phase of the project took much longer than originally anticipated. The interdisciplinary nature of the team was highly beneficial in allowing the team to use each member's area of expertise, while allowing the members to learn from each other and gain experience in areas not traditionally covered in their respective curriculums."

**Acknowledgements**

We would like to extend our thanks and gratitude to Dr. Gary Viviani of VivatronX and Mr. Cameron Rogers of the Texas A&M Aerospace Engineering Department. Without the support of these individuals we would not have been able to achieve the level of progress indicated in this report. Without Dr. Viviani's unique vision and expertise in the creation of autonomous systems, we could not have stayed true the vision of Railbot. Mr. Cameron Rogers has served as our graduate student advisor and his leadership and technical skill has assisted bringing the vision of

Railbot to life. It is to these individuals that we dedicate this paper.

**Bibliographic Information**

1. Lagoudas, Magdalini, and Jeffrey Froyd. "Multidisciplinary Vertically Integrated Teams Working on Grand Challenges." 2015 ASEE Annual Conference and Exposition Proceedings, June 2015. doi:10.18260/p.24515.
2. Valasek, John, and Kristi Shryock. "Enhancing Systems Engineering Content in Aerospace Courses: Capstone Design and Senior Technical Electives." 2015 ASEE Annual Conference and Exposition Proceedings, June 2015. doi:10.18260/p.24004.
3. Dr. Johnson, Suzette S. "Agile Systems Engineering: What is it and What Have We Learned?" Software Process Improvement Network. March 22, 2012. Accessed February 11, 2017. http://www.uces.csulb.edu/spin/media/pdf/INCOSE_LA_SPIN_AgileEngineering_Johnson.pdf.
4. Dr. Valasek, John. *Introduction to Systems Engineering*. PDF. College Station: Vehicle Systems & Control Laboratory, September 4, 2016.