Achieving a Culture of Software Quality for Android Robots

Vivek Kumar[1], Shreyes Joshi[2], Catherine Lee[3], Anya Bohra[4], Krish Aditya[5], Sneha Jeyasingh[6], and Bo Li[7]

totalchaosftc@gmail.com, bo.a.li@intel.com

Abstract

Robotics software quality is very crucial to control robot hardware while interacting with the physical world to accomplish a set of tasks. The software needs to tackle sensor errors, software exceptions, robot hardware failures and uncertainties from the external environment. This paper describes the challenges of achieving a culture of software quality for an Android platform robot based on the experiences of the First Tech Challenge (FTC) robotics team, Total Chaos.

To maximize the quality of FTC work, Charles Handy Athena culture model was employed to provide a small-team culture, working with flexibility, adaptability, and empowerment. The optimum level of power distributions and cooperation was established to maximize team efficiency. The use of the machine learning, embedded with modular software and hardware, further increased the effectiveness of the design process. The case study of an FTC Android robot demonstrates how the culture of teamwork and collaboration improved the development process, how modular design and documentation drastically improved the efficiency and quality of Android robot design and implementation, and how Deep Learning TensorFlow based Neural Network enabled accurate autonomous object detections.

Biography

The authors are high school students from the Total Chaos robotics team. The team is a four-year veteran FIRST Tech Challenge (FTC) team from Portland, Oregon. They have participated at the state and world level competitions. In 2016, they advanced to the world championships getting recognitions for their competitive robot, programming and contribution to the communities. The team developed a streamlined design and building process while ensuring software quality. The team also focuses on outreach activities in educating young students in communities about Science, Technology, Engineering and Mathematics (STEM) through Tech Talks, community open houses, and mentoring younger robotics teams.

Bo Li works in software developments and managements for Logic Technology Development in Technology and Manufacturing Group (TMG) at Intel. He joined Intel in 1999 after receiving Ph.D. from Georgia Institute of Technology. Since then, he has been focusing on software development, quality, performance, and testing for variety of software applications. Outside of work, he volunteers to mentor robotics teams in the community for students to foster passion and knowledge in STEM.

[1] Sunset High School, Portland, Oregon
[2] Sunset High School, Portland, Oregon
[3] Sunset High School, Portland, Oregon
[4] Westview High School, Portland, Oregon
[5] Jesuit High School, Portland, Oregon
[6] Westview High School, Portland, Oregon
[7] Logic Technology Development, Technology and Manufacturing Group, Intel Corporation

1 Introduction

It has long been recognized that experiential and hands-on education provides superior motivation for students to start at a young age. Robotics has been shown to be a superb tool for hands-on learning, not only of robotics itself, but of general topics in Science, Technology, Engineering, and Mathematics (STEM). Learning with robotics gives students an opportunity to engage with real life problems that require STEM knowledge. Mataric et al [1] describes the approach of enabling hands-on experiential robotics for all ages through the introduction of a robot programming workbook and robot test-bed. The workbook and test-bed provide readily accessible materials to K-12 students for direct immersion in hands-on robotics. Additionally, the success of learning science through robotics is explored by Chow et al [2] through their experiences.

For Inspiration and Recognition of Science and Technology (FIRST) is a team-based robotics program created for students aged 6-18. Within the FIRST program, there are four different sub-programs: Jr. FIRST Lego League (Jr. FLL) for students aged 6-9, FIRST Lego League (FLL) for students aged 9-13, FIRST Tech Challenge (FTC) for students aged 12-18, and FIRST Robotics Challenge (FRC) for students aged 14-18 [3]. As the programs progress, the tasks become more difficult and require deeper STEM knowledge to accomplish. Jr. FLL and FLL are Lego based programs, while FTC and FRC require teams to use Java programming and build with metal parts. Teams are required to develop strategy and build robots based on sound engineering principles. The ultimate goal of FIRST is to reach more young people with a lower-cost and more accessible opportunity to discover the excitement and rewards of STEM.

1.1 Background for FIRST Tech Challenge Robot Programming

In the FTC program, teams of 5-15 members build, program, and operate robots that compete in a head-to-head challenge. The field the robots compete on are 12ft by 12ft in size, and the robots have a size limit of 18in by 18in by 18in.

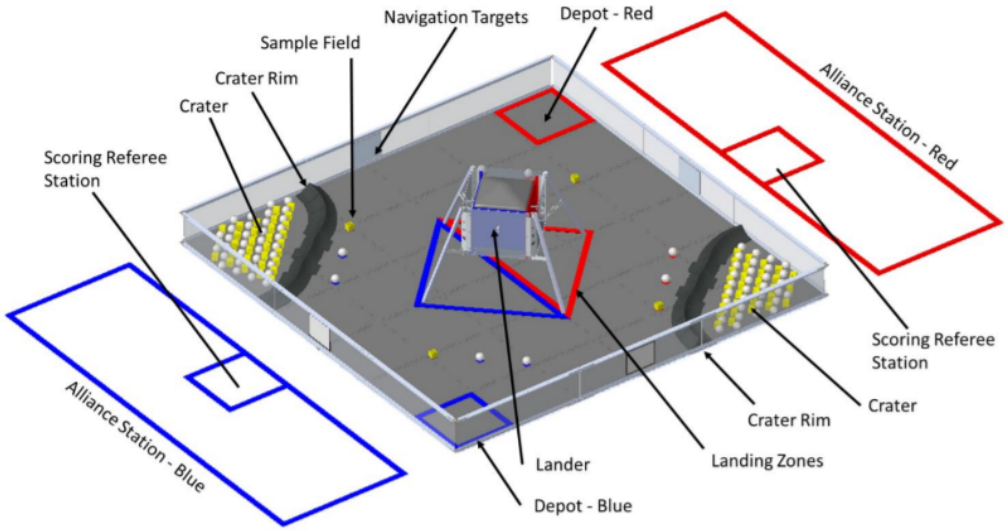


Figure 1. FTC field setup for 2018-19 season [4]

Figure 1 showcases a digital drawing of the field for the 2018-2019 Rover Ruckus FTC challenge. Although the challenges change every year, there are time-based rules that stay the same. All of the matches are 2 minutes and 30 seconds long. The first 30 seconds is the autonomous period where the robot is required to accomplish tasks autonomously. For the Rover Ruckus challenge, every robot starts in the parking zone or hung on the lander via the lander support bracket (See *Figure 2*). Driver control

period begins 5 seconds after the autonomous period ends. For the driver control period, each team designates two team members to be in charge of controlling the robot, and one coach who helps the drivers keep track of time and looks for ideal positions. End game period is the last 30 seconds of driver control time. During end game, a new set of challenges are open for the teams to complete. For the most part, the challenges during end game are the hardest and highest scoring. The more challenges the robots complete, the more points the team scores. The play style is 2 teams vs. 2 teams. Each team is paired up with another team to form an alliance. Two alliances compete against each other during every match. Alliances are chosen randomly and change from match to match. The alliance that accumulates the most points wins the match.

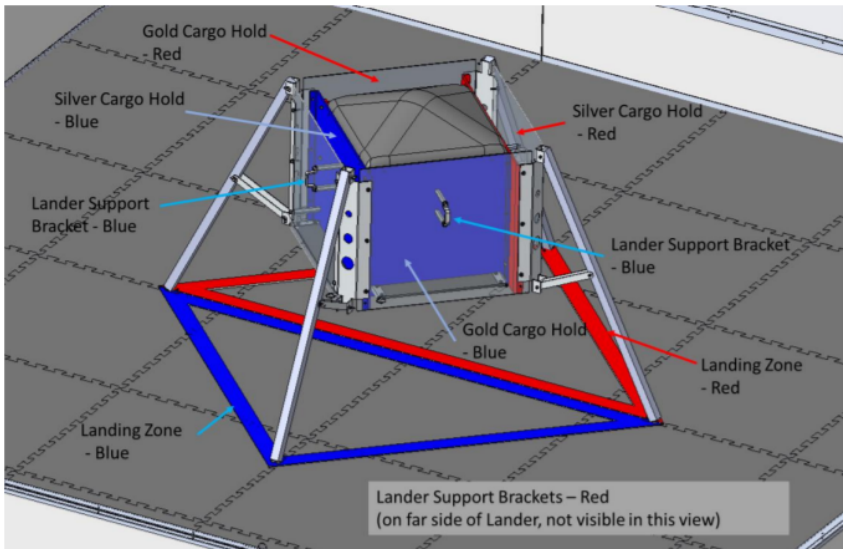


Figure 2. FTC Rover Ruckus Lander Details [4]

Table 1 shows the Rover Ruckus FTC challenge Game Period and Scores:

Table 1: 2018-2019 Rover Ruckus FTC Challenge Game Period and Scores

Game Period	**Scoring Objectives**
A— Autonomous Period	● Drop down and unlatch from the lander (30 points) ● Knock off only the gold mineral from the tape mark (25 points) ● Drop a small team marker in their corresponding depot (15 points) ● Park in a parking zone (10 points)
B— Driver Control Period Scoring	● Collect minerals from the crater and deposit them in the lander (5 points each). The minerals have to be deposited in the corresponding cargo hold to be counted for points. The robot can only hold 2 minerals at a time. ● Collect minerals and deposit them in the depot (2 points each)
C— End Game Scoring	● Continue scoring minerals (5 points each) ● Latch and hang on the lander (50 points). ● Partially park in crater (15 points) ● Fully park in crater (25 points).

Figure 3. The control system of an FTC robot and its subsystems consisting of joysticks (used by drivers), Android smartphones with pre-loaded software programs, and core components (e.g., external sensors, motors, power, etc.) to detect and control FTC robot [5]

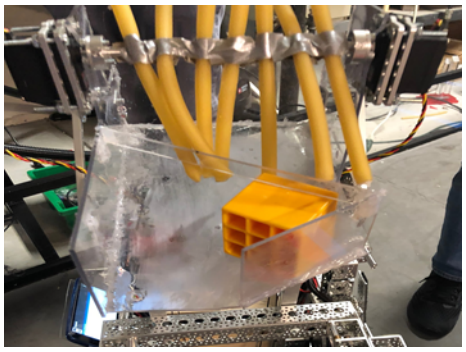
FTC utilizes Java through the Android Studio platform to program the robot to move and compete (See Figure 3). The Android programs are downloaded and run on a phone which is called the robot controller. The robot controller is wired to the Rev Expansion hub, which is connected to the motors and servos. Through this, the robot controller can tell the module when to give servos power. The robot controller also uses Wifi direct to communicate with the Driver Station, which is a phone that is kept near the user. The Driver Station acts almost like a remote control, where certain programs can be selected and be told to run. Gamepads are connected to the driver station through a USB hub. When a driver clicks a button on the gamepad, it registers with the driver station, which sends a signal to the robot controller through Wifi direct. Using the downloaded program, the Robot Controller tells the Expansion hub what motors or servos to give power to, making the robot move.

1.2 Introduction to FTC Robot for Rover Ruckus Challenge

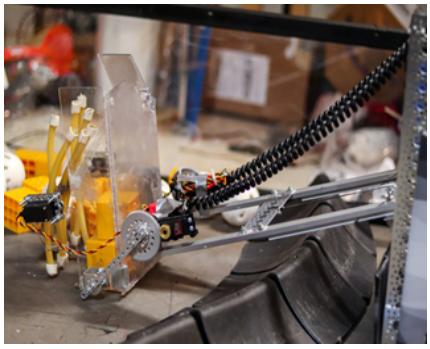
The robot design is the foundation to solve the challenge for the FTC season. For 2018-2019 Rover Ruckus challenge, the robot drivetrain from Total Chaos team is a four wheel drive with mecanum wheels for added maneuvering ability, including sideways and diagonal movement. The wheels are powered by NeveRest 40 motors mounted vertically, attached by bevel gears in each of the corners of the robot. This is advantageous as it empties out the center of the robot, which not only provides more space for other robot modules, but also enables easier access when making repairs and troubleshooting issues. The lower central space of the robot features the REV electronic interface modules while the upper side holds the Moto G4 Play robot controller to aid with Vuforia. The entirety of the robot was built to produce an efficient scoring system that had the ability to stand its ground.

A major part of the robot is the mineral scoring module, which was used to retrieve and score gold and silver minerals during the tele-op period. The mineral scorer takes up a large portion of the robot, due to its overwhelming value in this year's game as a major differentiator amongst top teams. The scoring mechanism utilizes two sets of linear slides that can extend 35 inches, which allows to collect minerals by extending over the crater walls. The whole mechanism is attached to two motors which allows it to pivot up and down to score minerals into the lander.

The box can hold two minerals and has the ability to sort the minerals. It is observed that gold minerals have a height of 2 inches while silver minerals have a height of 2.75 inches. Using the measurements of the minerals, a box was created where the gold and silver minerals go in opposite directions, into their corresponding cargo holds. The addition of a sorter eliminates the need to sort through the crater and collect only silver minerals or only gold minerals (See Figure 4).



(a)



(b)

Figure 4. Total Chaos FTC Robot a) Box b) Depositing Mechanisms

Three sensors were utilized to maximize the efficiency during both the tele-op and autonomous (Figure 5).

1) The motor encoders, which are sensors built into each NeverRest motor that use counts to track the rotations of each motor. By using the diameter of the wheel and a counts per rotation constant specific to the motor, the robot is able to convert inches into counts. This meant that the software can tell the robot to move x inches and it would. However, these encoders aren't entirely accurate. While this doesn't cause too many problems with straight movements, the issues really appear with turns, which are more dependent on accuracy. Because of the inconsistency, it was decided to use the REV IMU gyro sensor for turns. This sensor was built into the Rev power distribution module, which gave power to all of our motors.
2) The gyro tracked the angle of the robot, and allowed the robot to make turns to specific angles, instead of depending on encoder counts.
3) The final sensor that was used is the phone camera, which allowed the robot to locate the gold and silver minerals, and knock the correct one.

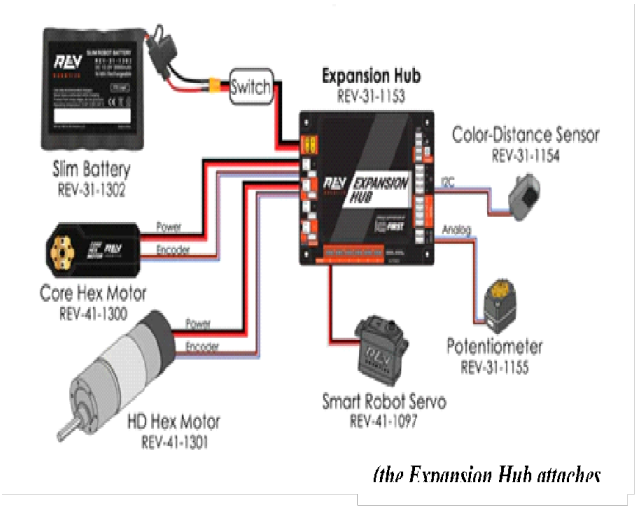


Figure 5. Total Chaos FTC Robot Sensors

1.3 Introduction to a Culture of Software Quality

FTC software quality is very crucial due to the high expectations of consistent performance during FTC competitions. If the software fails, it can significantly impact FTC competition results. A quality-focused culture leads to high quality products, which allow for high scoring capabilities during competitions.

This paper presents two main challenges in accomplishing a culture of software quality for FTC robotics competitions. The first challenge is ensuring that the software can be adapted rapidly while ensuring high software quality. Rover Ruckus was released on September 8th, 2018, and the first competition was on January 27th, 2019. This means we had a short four months to design, build, test, program and refine a competition-ready robot. The second challenge is ensuring that the robot can utilize machine learning software to minimize the effects of the constantly changing physical world and make the right decisions in different situations. The machine learning algorithms have to be tested thoroughly to ensure high quality robot performance.

2. Achieving a Culture of Software Quality in Android Robots

With a small team of six members, efficiency was crucial to produce a functional and competitive robot within the 4 month build season. As our team gained experience, we refined our work ethic to complete all of our tasks in a timely manner. We found it necessary to develop a structure that would utilize the strengths and interests of every team member. We've found that the Athena Culture of Charles Handy's management theory has helped us accomplish those goals best. Along with the Athena Culture, we also utilized modular design, effective documentation and Tensor Flow machine learning to further increase the quality of our robot.

Charles Handy is an author and philosopher known for his work in organizational behavior and mechanics. He is also known for developing the Charles Handy Theory, in which he describes four different management structures - based on power, role, task, and person (See Figure 6 [5]). Each of these structures is commonly connected to four Greek Gods: Zeus (power), Apollo (role), Athena (task), and Dionysus (person). The Athena culture is a task-based culture that centers around flexibility and adaptability. The core of this structure involves subteams that are formed to complete different tasks. By having subteams, individuals are enabled to work on a myriad of projects based on their own skills and interests. It also saves time because tasks can be done in parallel [6].

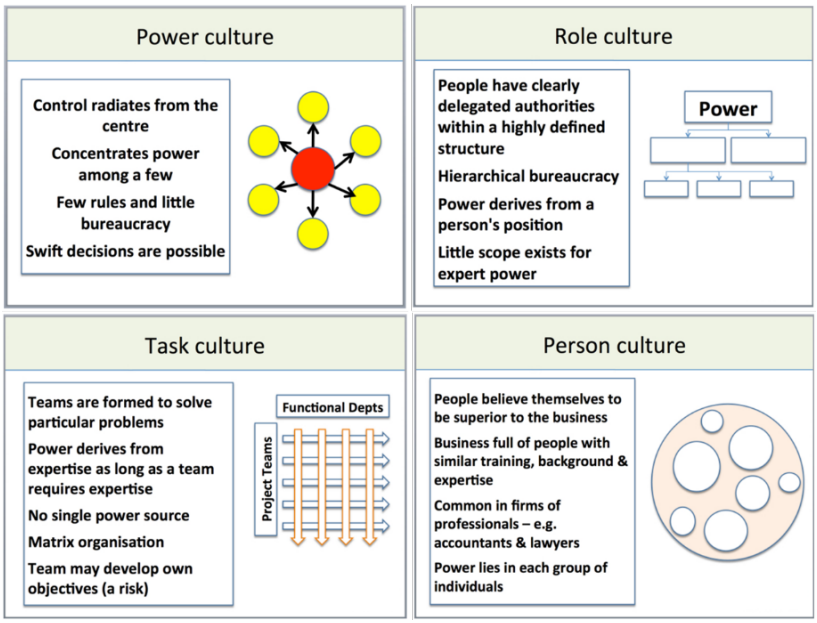


Figure 6. Charles Handy Model of Team Management

We also utilized modular design to allow changes to certain elements of the robot to be fast and efficient. The independent nature of each mechanism allowed us to isolate different parts of the robot to easily pinpoint a problem. After pinpointing a problem, we could easily remove the mechanism and make changes without altering the rest of the robot.

Performance expectations for the robot increase after every elimination tournament, meaning that constant improvements need to be made. The use of effective documentation allowed us to note past robot performance, previous iterations, stress tests, and test runs to more easily brainstorm and create the next design.

Lastly, we employed TensorFlow machine learning technology for software quality improvements. TensorFlow is a deep learning library recently open-sourced by Google [7]. TensorFlow provides primitives for defining functions on tensors and automatically computing their derivatives. Formally, tensors are multilinear maps from vector spaces to the real numbers and fundamental mathematical constructs in fields such as physics and engineering. Historically however, tensors have made fewer inroads in computer science, which has traditionally been more associated with discrete mathematics and logic. The state of affairs has started to change significantly with the advent of machine learning and its foundation on continuous, vectorial mathematics. Modern machine learning is founded upon the manipulation of tensors and the calculus of tensors. The TensorFlow Machine Learning process utilizes image recognition (Vuforia) to record the location of the minerals for later use. When hanging, the robot phone camera activates Vuforia, allowing the TensorFlow image recognition to initialize and scan the field for objects. The TensorFlow Machine Learning system is trained to recognize the gold/silver mineral by being shown numerous images of the objects from multiple angles, allowing for a complete perspective to be produced for the high quality of object detection software.

3. Case Study: FTC Android Robot Culture of Software Quality

3.1 Charles Handy's Athena Culture (Task Based Work Culture)

The Athena culture is a team-based culture that centers around flexibility and adaptability. The core of this structure involves subteams that are formed to complete specific tasks. Additionally, the Athena culture is flexible; individuals are enabled to work with several subteams based on their own skills and interests to accomplish different tasks and missions.

As described in sections 1.1, 1.2 and 1.3, the Rover Ruckus challenge requires robots to accomplish a multitude of varying tasks. One robot must have several subassemblies to accomplish them all. We created several subteams for each subassembly, in addition to programming and outreach (See Figure 7). Following the Athena Culture team management enabled our team to design, build, and program every subassembly in parallel and accomplish the robot control tasks to meet the critical need dates.

By utilizing the Athena culture, we were able to drastically decrease our overall build, programming and testing time. Without the sub-teams system, it would've taken an estimated 12 months to complete our robot (See Figure 8). Being able to work on many tasks in parallel allowed us to completely design, build, program, and test our robot in 4 months. Furthermore, software development was accelerated as we were able to create and validate code for each assembly without it being on the completed robot. When all subassemblies were implemented onto the robot, we already had reliable code that was developed alongside the hardware design and implementations.

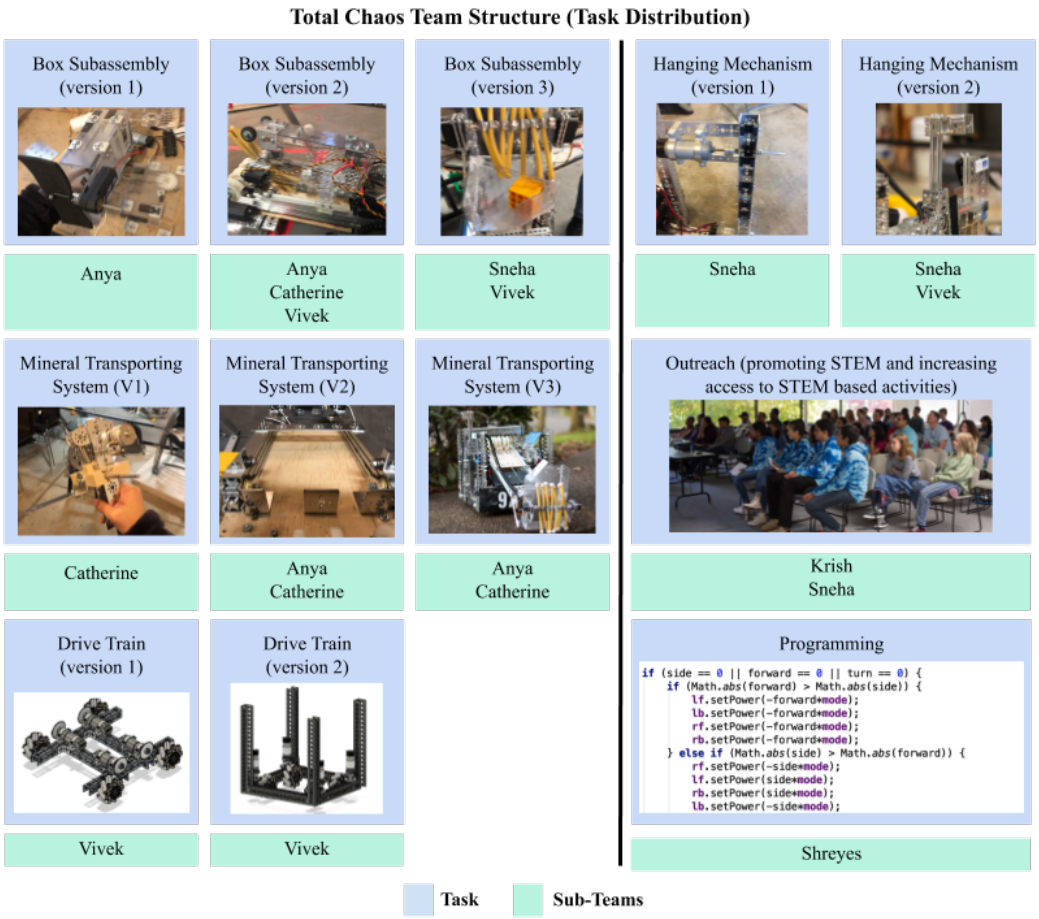


Figure 7. FTC Robot Subteams for Each Subassembly

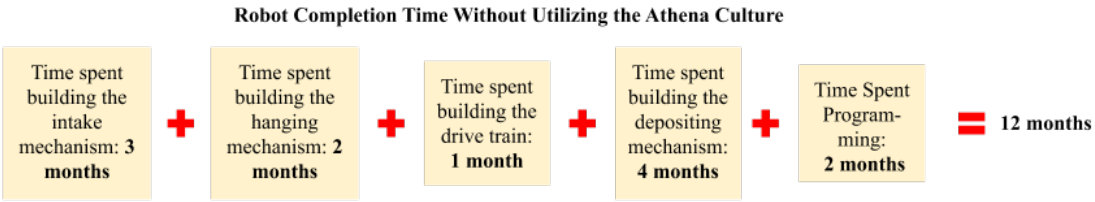
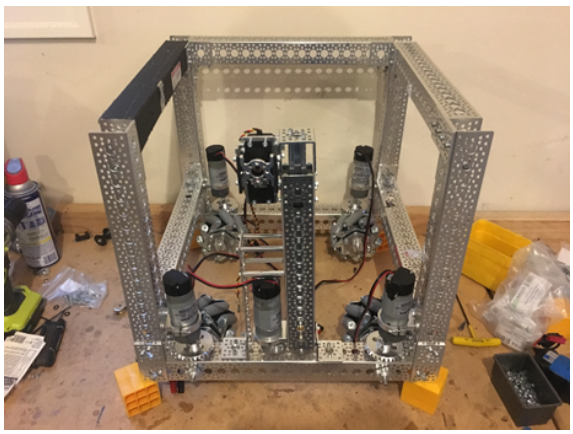


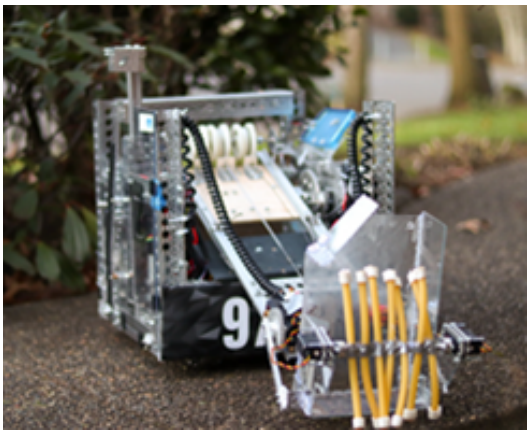
Figure 8. FTC Robot Completion Time without Utilizing the Athena Culture

3.1.1 Modular Designs for Charles Handy's Athena Culture

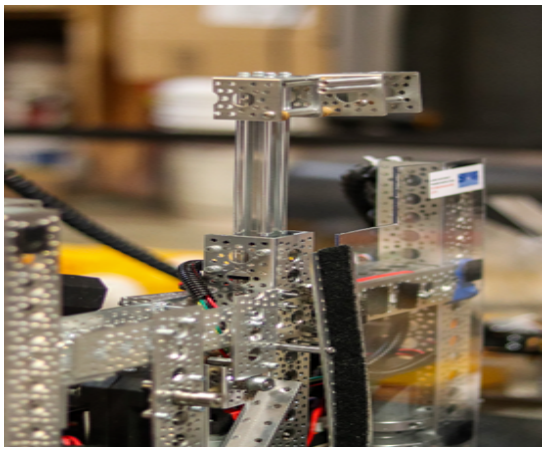
We split our robot into four separate modules: 1) box subassembly 2) mineral transporting system, 3) hanging mechanism and 4) drive train.



(a)



(b)



(c)

Figure 9. Total Chaos FTC Robot (a) Drive Train (b) Mineral Intake Mechanism (c) Hanging Mechanism

After employing the Athena culture within our team, we decided to split our robot into separate modules for each subteam to focus on (See Figure 9). The independent nature of the modules proved to be beneficial at many points during the season. After our first competition in January 2019, we began preparing for the state competition in March 2019. We wanted to increase our scoring capability by fixing the drive train -- at our first competition, the robot began swerving left instead of driving straight-- and rebuilding our pivoting platform to decrease the weight. When rebuilding the platform, we were able to easily remove the mechanism from our robot and rebuild it without implicating the rest of the robot. When we finished building and testing the new mechanism, the installation process took a mere 20 minutes. The ease of this process was only made possible because of the separate modules that we utilized.

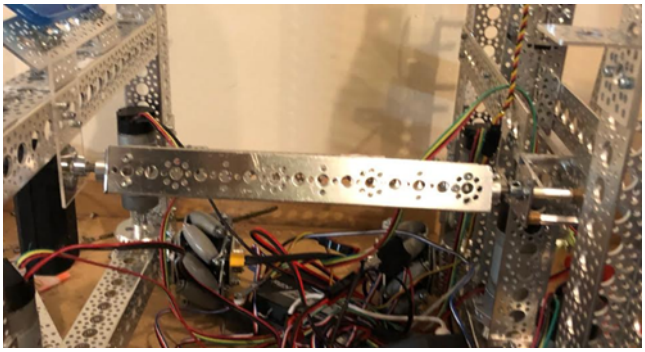
Before reducing the weight of the platform by 1.25 lbs, it required more torque to rotate the platform from intake to scoring position. As a result, the motors on the platform would periodically stall because they could not provide the necessary torque to rotate the mechanism. When the motors stalled, the gears would slip due to the weight of the platform forcing the gear in motion without any motor power. As a result, the slippage created low rotational accuracy, so we were unable to use the motor's encoder counts to track the position of the motor. After decreasing the weight of the platform, the encoder counts became reliable, so we were able to utilize them for driver control enhancements during tele-op. For example, we were able to prevent the box on the platform from slamming against the ground when rotating to collecting position. If the encoder counts were between 125 and 0, we knew the platform was between around 30 degrees of its starting position, so we stopped giving power to the motors and allowed gravity to slowly bring the mechanism down (See Figure 10).

```
if (gamepad2.right_trigger > 0 && rotateLeft.getCurrentPosition() > 300) {
    rotateLeft.setPower(0);
    rotateRight.setPower(0);
}else if (gamepad2.right_trigger > 0) {
    rotateLeft.setPower(gamepad2.right_trigger);
    rotateRight.setPower(gamepad2.right_trigger);
} else if (gamepad2.left_trigger > 0 && rotateLeft.getCurrentPosition() < 125) {
    rotateLeft.setPower(0);
    rotateRight.setPower(0);
} else if (gamepad2.left_trigger > 0) {
    rotateLeft.setPower(-gamepad2.left_trigger);
    rotateRight.setPower(-gamepad2.left_trigger);
} else {
    rotateLeft.setPower(0);
    rotateRight.setPower(0);
}
```

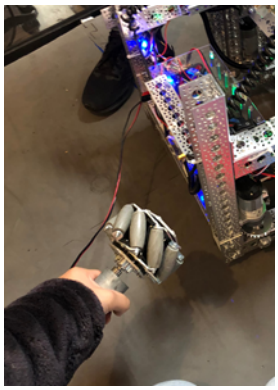
If the position of the left motor that powers the platform is less than 125 encoder counts, then set the power of the left and right motors to zero

Figure 10. Software Sample to Control the Platform When Rotating to Collecting Position

At our first competition, the autonomous encountered a drivetrain swerving issue. The modular nature of our robot allowed us to test the drivetrain motors while rebuilding the pivoting platform because the two mechanisms were completely independent (See Figure 11). After discovering the motors attached to the wheels were still functional, we realized the swerving issue stemmed from the wheels; many of the rollers on the mecanum wheels were too tight and could no longer rotate freely. After loosening the rollers, we conducted 10 tests and the autonomous program worked perfectly.



(a)



(b)

Figure 11. Total Chaos FTC Robot Drivetrain Swerving (a) Drivetrain (b) wheel

3.1.2 Effective Documentations for Charles Handy's Athena Model

A practical and easy method to maintaining quality and consistency over the course of the season lies within the engineering notebook [8]. Our notebook contained thorough technical descriptions of the robot design, in addition to meeting notes that described our journey in designing the robot. Throughout the season, we made sure that we thoroughly documented every single meeting. Any changes made to the robot, code, or outreach activity planning is described in detail, alongside reasoning behind the change. Testing procedures and results were also documented in full detail, giving us a reliable source of data from which we could evaluate the strengths and weaknesses of our subassemblies or program. In the latest Rover Ruckus season, documenting every program iteration proved helpful during many stages of the programming process. At the beginning of the season, we planned and documented the ideal path the robot should take during autonomous. Having a reference as to where the robot needed to be made programming far easier; whenever we finished programming a section of the path, we knew exactly what to program next.

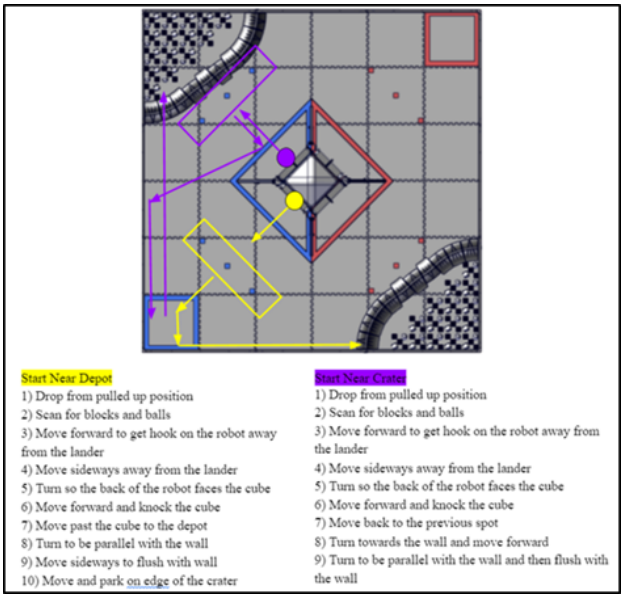
Documenting the steps to solve a problem was another effective practice (See Figure 12.a). When coding the robot, issues sometimes resurfaced even after we had attempted to solve it. When our continuous servo motors would not function as we wanted them to, we were able to refer to meeting notes that detailed a similar issue. By referencing how we solved the problem last time, we were able to build upon the previous solution and solve the issue quickly.

One of the most important documentation practices was recording our goals and criteria (See Figure 12.b). Whenever we began a new subassembly or a new program, we made sure to thoroughly document our goals, our timeline, and our criteria for the finished product. When we set these goals on paper, we are able to orient ourselves to these high-level goals throughout the season, always being able to refer to an agreed-upon list of requirements and ideas.

Session # 14	2/15/2019

Objectives	Reflections
Fixing issues with continuous servos	• We were having issues with setting the power of the continuous servo • It would only set power when we took readings from the joystick • If we tried to set the power to one the servo wouldn't move at all • When the joystick controlled the power it would move inaccurately, sometimes very slowly • To solve the problem we simply tried giving it an double number slowly lowered the power • Once the power reached 0.8 the servo started moving at full speed • The power was constant, and did not slow down like I did before

(a)



(b)

Figure 12. Total Chaos FTC Robot Documentation Examples: (a) Steps to Solve Servo Problems (b) Autonomous Period Goals and Criteria

3.2 TensorFlow Machine Learning for Autonomous Phases

We have two different autonomous combinations to complement our alliance partners. The diagram below shows both paths we take (See Figure 13), which consistently scores all 80 points.

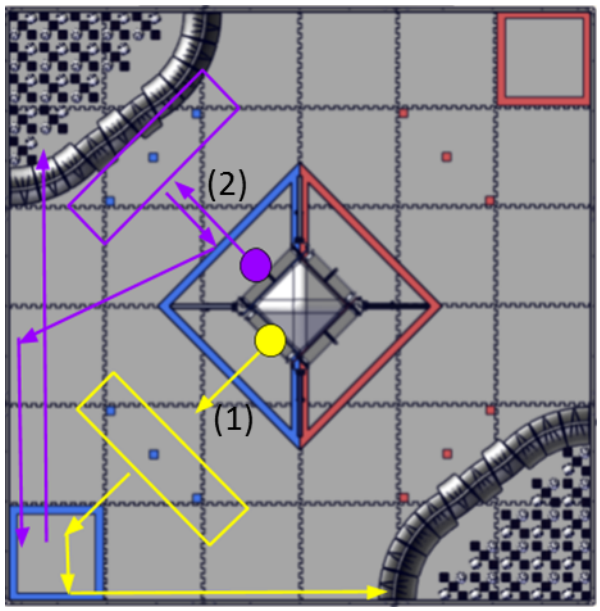


Figure 13. Illustration of Total Chaos Autonomous Robot Movements

Table 2: Rover Ruckus FTC Challenge Autonomous Robot Action Sequence Example

Path (1): Near the Lander	**Path (2): Near the Crater**
1) Drop from pulled up position 2) Scan for blocks and balls 3) Move forward to get hook on the robot away from the lander 4) Move sideways away from the lander 5) Turn so the back of the robot faces the cube 6) Move forward and knock the cube 7) Move past the cube to the depot 8) Turn to be parallel with the wall 9) Move sideways to flush with wall 10) Move and park on the edge of the crater	Drop from pulled up position 2) Scan for blocks and balls 3) Move forward to get hook on the robot away from the lander 4) Move sideways away from the lander 5) Turn so the back of the robot faces the cube 6) Move forward and knock the cube 7) Move back to the previous spot 8) Turn towards the wall and move forward 9) Turn to be parallel with the wall and then flush with the wall 10) Move to the depot and drop the marker 11) Move and park on the edge of the crater

One important part of the autonomous objectives is to identify the location of the gold mineral accurately. We utilized Vuforia software and Google's TensorFlow to locate the position of the two silver balls and the gold cube. Our initial idea for finding the location of the gold mineral was to use a long metal piece with a color sensor on it and attach the piece to a servo. The mechanism would sense the color of the gold cube and knock it by rotating the servo. We decided to use the TensorFlow software instead because it is more reliable. Additionally we saved a lot of time by doing a quick scan using the camera instead of moving close to every mineral to scan for the color.

The TensorFlow Machine Learning [9] process utilized image recognition (Vuforia) to record the location of the minerals for later use. When hanging, the robot phone camera activated Vuforia, allowing the TensorFlow image recognition to initialize and scan the field for objects. The Machine Learning system has been trained to recognize the gold/silver mineral by being shown numerous images of this object from multiple angles, allowing for a complete perspective to be produced. Once the objects are visually scanned, it compares previously learned images to the real life situation. Additionally, every iteration of the scanned objects are added to a cloud recognition database to enhance future accuracy of scanning. In some scenarios, some images are related with the object tucked behind another object, so the TensorFlow can recognize the gold/silver mineral with only a partial image. After recognizing the location of the gold/silver mineral, the camera utilizes the xy plot of a Cartesian plane to generate an exact location for gold/silver mineral. Based on previous testing, we were able to narrow down the expected ranges of the gold/silver mineral, then allowing us to locate the xy location. With the paths for each mineral programmed, we are ready to begin detecting the mineral arrangement. We wanted to use it so we could recognize which position the gold mineral was in, and tell the robot to push that mineral away accordingly.

Furthermore, when we mounted the phone and started the test program, we found that when the phone was in an upright, portrait position, the phone camera was not wide enough to see all three minerals. Switching to landscape allowed the phone to see the whole arrangement. After adjusting the setup, the camera was positioned appropriately in order to compare the images in the right directions. For example, when the robot is hanging, the phone camera will have to be angled down to see the arrangement. It is very important to have the right inputs to the machine learning algorithm (e.g., the positioning and angle of the phone mount.). With the advanced troubleshooting practice and methodology, we can detect the right mineral 100% of the time (See Figure 14)

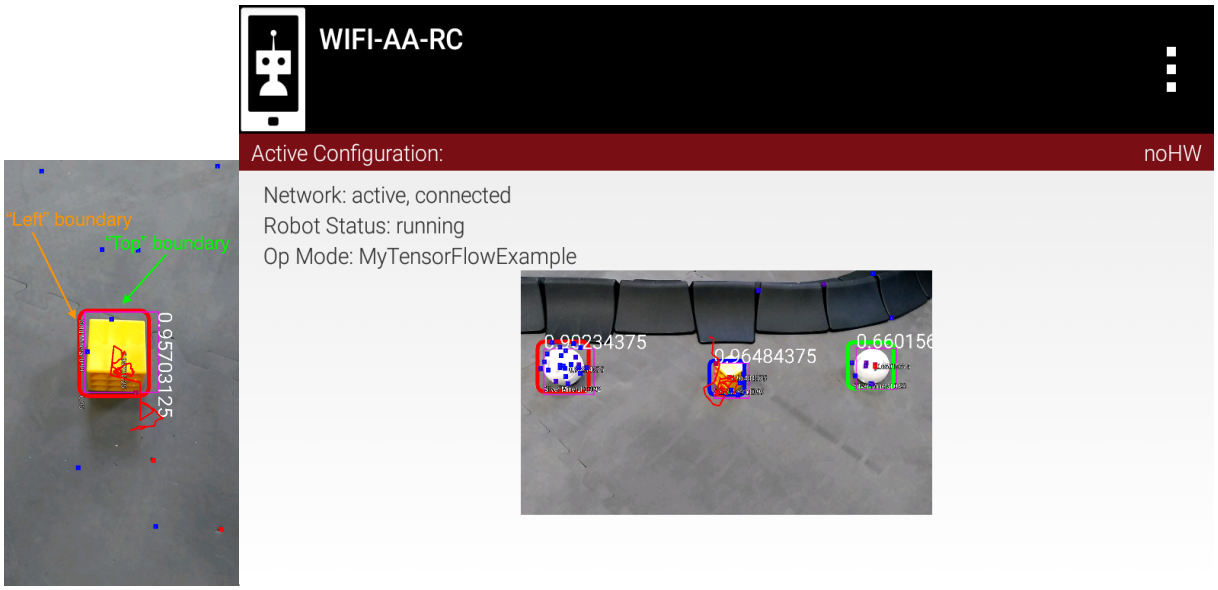


Figure 14. Illustration of Golden Mineral Detections Using TensorFlow Machine Learning [10]

4 Summary

This paper presented our approaches to achieve a culture of software quality in Android robots that integrates robot software and physical worlds together. Case studies were shared to utilize Charles Handy's Athena Culture model to improve software quality in Android robots for the robot performance improvements in FTC robot competitions. The detail-oriented team structure, effective document, and

modular designs were presented to take into considerations of multiple factors including robot design, build, programming and team coordination. With the evolvement of robot performances and capabilities, the software quality was improved. It was proved that the software quality Charles Handy's model is a very powerful tool to guide the team to close software quality gaps in critical areas for consistent robot performances. Furthermore, the paper also presented TensorFlow Machine Learning to improve the software quality in autonomous phase to minimize dependencies on the external physical sensor readings. All of these methods can be utilized in industries to improve software quality when handling the physical world uncertainties and complicities.

Acknowledgments

The authors wish to thank the reviewer Joseph Ruskiewicz for his invaluable assistance and discussions on the culture of software quality. In addition, the authors also wish to thank the Oregon Robotics Tournament & Outreach Program (ORTOP) whose assistance is very crucial for the success of FTC events in Oregon.

References

[1] Mataric, Maja J., Koenig, Nathan and Feil-Seifer, David. 2007. "Materials for Enabling Hands-On Robotics and STEM Education", *AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education, Stanford, CA.*

[2] Chow, Kingsum, Chow, Ida, Niu, Vicki, Takla, Ethan and Brillhart, Danny. 2010. "Software Quality Assurance in the Physical World", *Proceedings of the Pacific Northwest Software Quality Conference. Portland, Oregon. USA.*

[3] FIRST Homepage: http://www.usfirst.org/ (accessed July 18, 2019).

[4] FIRST, 2018-2019 Game Manual: https://firstinspiresst01.blob.core.windows.net/ftc/2019/gemf2.pdf (accessed July 18, 2019).

[5] Tutor2u. Handy's Model of Organisational Culture. https://www.tutor2u.net/business/reference/models-of-organisational-culture-handy (accessed July 18, 2019).

[6] Dininni, Jeanne. 2011. "Management Theory of Charles Handy". https://www.business.com/articles/management-theory-of-charles-handy/ (3.1 Athena Culture) (accessed July 18, 2019).

[7] Qualcomm. 2019. "2019-2020 FTC Android Studio Tutorial". https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/android-studio-tutorial.pdf (accessed July 18, 2019).

[8] Qualcomm. 2019. "2019-2020 FTC Wiring Guide". https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/robot-wiring-guide.pdf (accessed July 18, 2019).

[9] "TensorFlow for Deep Learning". https://www.matroid.com/dlwithtf/chap1-2.pdf (accessed July 18, 2019).

[10] FTC Engineering. "Java Sample TensorFlow Object Detection Op Mode". https://github.com/ftctechnh/ftc_app/wiki/Java-Sample-TensorFlow-Object-Detection-Op-Mode (accessed July 18, 2019).