Educating a Team from a Leadership Position: Encuraging and Facilitating the Growth of Knowledge and Expeierience in Younger Members
Isaac Yates
Norman Advanced Robotics

Educating a Team from a Leadership Position: Encuraging and Facilitating the Growth of Knowledge and Expeierience in Younger Members

1 A Backstory of NAR and my Role on the Team

Nicky Halterman and I sat together in the middle of the botball room, discussing how to write camera code on a CBC. We were freshmen, excited and eager to create. Nicky would leave the team in a year in order to dedicate his time to debate and soccer, others would follow for similar reasons, and slowly the class of 2015 dwindled down to two people: myself and one other senior.

During that moment with Nicky, our sponsor, David Askey, interrupted us briefly. He said:

"You guys are going to be leading this thing in four years, right?"

In 2012 Norman Advanced Robotics placed 2nd in the world at the Global Conference for Educational Robotics. In 2013 NAR also placed 2nd. Flash forward to 2014, and the team was clearly struggling, riding on the talented shoulders of about 3 seniors and myself (a junior). Two weeks before the tournament, a completely new robot was built and programmed. It failed to deliver the score we had aimed for (no surprise). During GCER 2014 Norman Advanced Robotics scored lower overall then it had in several years. The extremely concentrated talent from a year before had all graduated, and to make matters worse the team was still top heavy. Coming into 2015, there were four seniors on the team. NAR lost two of them during the next fall. Our freshmen class was as large as our sophomore, junior, and senior classes combined.

1.1 Isaac Newton, Teamwork, and NAR's Failure

I think the quote "standing on the shoulders of giants" has a little different meaning than the situation stated above. But I have a broad question: Was Newton the only person capable of doing math? No! There were dozens, if not hundreds or thousands of young budding scientists working on less advanced, but still just as important concepts as calculus, not to mention the written work of Greeks, Romans, Arabs, Chinese, and more recently dead scientists such as Galileo Galilei. Where would we be today if nobody after Newton continued his work? Modern Calculus has far more uses than physics, which was what Newton primarily used it for.

A team is the same way. A team is not made up of a couple talented leaders. A team is made up of them, plus those that came before them, plus those that come after. NAR had done something wrong. We weren't meeting and beating the accomplishments from previous years. Yes, there were plenty of shoulders stand on, but nobody knew how to climb.

When I became leader of the team, I had this thought in the back of mind. I knew that we needed to climb, not just stand, atop the work of alumni, and I was aware what the current freshmen would have to do the same after Micheal and I graduated. Here is what I've done in the past year in an attempt to reach the goal of educating our younger members and providing them not just an easily accessible body of work on which to use for their purposes, but also the right gear and know-how to use that work, and to bring the team back together.

2 Writing down Ideas is Important

As a leader, it is your first job is to do more work than the rest of the team. Your second job is to lend the team direction. From the moment I stepped onto LAX in California after GCER 2014, I was reflecting on the year and how to make 2015 better. I opened a note app on my phone and wrote down ideas that came to me, because I knew that I would forget them by the time we started meetings back up in the fall. Then I told my team members at GCER to do the same. I did not use every idea I had, but here is a list of what I wrote down.

2.1 The List of Tips

- Have a camera man/woman who sole job is to take pictures of robots and jot down notes about them during different stages of development. This was hard to implement because most people who join bot ball want to build robots, not take pictures. Eventually, during the summer season, I managed to get a couple of the documenters on our team to set themselves a daily quota of pictures.
- A programming function board. This was supposed to be a white board where builders and programmers collaborated in specifying the sort of functions a robot would need while the robot is still being created. This would theoretically cut down on programming time. I have found that I cannot count on the team members using the function board unless I set aside time each meeting and get them to talk to each other.
- Periodic group conversations involving the whole team. Without communication, there is no team. Usually I would direct these at the beginning of every meeting. I would ask each member what their goal was for the day, and what they needed help with in order to achieve it. You would be surprised how often high schoolers have no idea what they are doing, but are too afraid to ask the team or even team leaders for help. Also, talking about difficulties daily as a group makes it easy to reassign people to other projects where they are more needed (or might have more fun!).
- Making a list of all game board parts that are likely to have inconsistent dimensions during tournament and putting the list where the whole team will see it during meetings.
- Draw out a large table with all the combinations of points possible on the game board labeled with the amount of points per combination. This takes a lot of work to do, but is worth it. Another thing you can do in addition is to mark the level of difficulty on a scale of your choice for each point combination.
- Have deadlines for building and programming as well as documentation. The robots will be modified and built upon constantly throughout the season, but setting a deadline for when the robot meets minimum requirements to do its job helps builders and programmers alike to speed up the process and be ready for the tournament. The robots will never be perfect, and at some point the massive rebuilds and re-prototyping has to stop so that the team can move on to the next step of development.
- It is often much easier to deny the enemy team points than to outscore them.
- Robots that work independently of one another are more likely to get their task done. A robot that relies on the performance of another robot messes up when the other robot messes up.
- Have parties. They are a good way to transition from one phase of the bot ball season to another, and people need the mental break.
- Use dropbox, Github, or some other file sharing website to organize the work of the programers. I used dropbox because it is simple and has virtually no learning curb for a group of computer savvy highschoolers. Make sure to have everybody put their name and the date of last update to each header file they write, that way the lead programmer can tell who to go to when run-time errors occur.

- Make a list of hardware concepts. Even if builders don't understand how to make them, at least they will have a starting point to solve engineering problems. Here's a list I made up with the help of NAR's lead builder and a couple juniors (we speak in NAR jargon a bit).
    1. Paralleling Arm
    2. Elevator Arm (use rubber wheels or pulley system)
    3. micro-servo mounts
    4. Triangles are stronger than squares
    5. Triangulating rubber bands.
    6. Single tribble catcher
    7. jank basket
    8. torque and big arms
    9. counter weights to solve torque problems
    10. hard aligning
    11. drive train designs
    12. tribble sorter
- Always strive for consistency, not maximum points. This becomes a problem on a team that is allowed the opportunity to watch older members build robots that consistently score about as much points possible. NAR has a good track history. Coincidentally, on one of the walls in NAR's bot ball room are the words: "complex flashing lights draw onlookers attentions, but consistency wins the tournament."
- Have a team coding library, and have at least one committed programmer, preferably the lead programmer, keep the library updated and running smoothly. In another paper, I talk about my own journey as lead programmer to completely redesign the drivelib of opencode, the tattered old lib that NAR had been relying on for four years.

3 Cleverly Spent Time = Success

So, now that I had ideas written down (and floating in my head), and now that GCER was well over and I was on my way home, what was the next step to rebuilding the team? I have a motto, based on discussions with past members of the team. Cleverly spent time equals success.

Following my motto, I began the earliest season start up in NAR history, according to our sponsor. I let August and October pass, and then in September we sent the emails out and searched the hallways of the high school for the lost freshmen who had done botball in middle school. Starting the year, I had made the analysis that we needed to train new programmers. I was the most experienced programmer on the team by far, and that was unacceptable. Cleverly spent time = success, so in late September I got together with our sponsor and set up the dates for a fall programming and building workshop, followed closely by the 3rd annual NAR mini-game.

3.1 The Mini-Game (History)

When I was a junior on the team, I spent a lot of time on the sidelines, because although the team was full of highly talented engineers and programers, it was not organized in a way that utilized all members, top to bottom. During my sophomore summer, Daniel Goree and Jeff Terry, leaders of the team at the time, found a smaller board in the botball room that nobody was using, and decided to mount it on a nearly broken Foosball table. Thus began the mini-game. I told several of the freshmen that if they could beat me in a mini robotics tournament, using that board and various points Daniel and I set up on it, then I'd give them five dollars. The effect of my challenge surprised the whole team, because previously lazy freshmen immediately began building a robot. In the end, none of the robots worked consistently by the mini-game tournament, sadly.

Here is what Daniel Goree had to say about the first mini-game when I asked him:

> "Norman Advanced was an extremely top-heavy team my senior year. Due to a variety of circumstances, my class had been the leaders of our teams since the seventh grade. By our final year, we had about eight seniors who had everything one could wish for in Botball members: natural talent in given specialties, hours upon hours refining those skills, and a combined 50 years of experience that yielded a deep understanding of the game and what it took to perform well. Going into the year, we knew it would be a waste not to utilize our tremendous resources, but with only three or four robots involved in our strategy, only a small handful of underclassmen would receive meaningful hands-on experience with the robots. On a team of twenty students, this was unacceptable. This tension lead to our creation of the mini-game. We assembled a small botball-like game table with our extra board materials and crafted a few simple rules that would serve as a game document. The hope was to provide an opportunity for our underclassmen to experience the entire Botball cycle from strategy formation to an in house tournament near the end of the season as the heads of their own small teams. When the start of the season finally arrived, we gave our annual basic programming tutorials, explained the mini-game, broke out the old processors, and let the underclassmen have at it. Regrettably, the first mini-game worked as well as bungee cord stopping a semi truck. A few students took advantage of it, however there were not enough to justify the tournament and no one had working robots by the deadline anyway. It was not until after we graduated that the mini-game began to effectively train younger members."

To sum up what Daniel said, the first NAR mini-game was a good expierence, but didn't exactly effect the underclassmen the way we wanted it to.

The second mini-game went about the same way, a half year later, and really didn't pick up speed. This was partially my fault. During 2014 I was much more involved in the team in general, and was distracted by regionals and globals.

End of History

3.2 The Third Mini-Game, 2014-15

The third mini-game was much different from the rest, because I made the conscious decision as the leader of the team to use it as a training tool for the WHOLE team. It gave me time to work individually with the team members, and it gave the team members, freshmen through juniors (there were only two seniors, and both of us were judges) time to build up their skills. If it had been an actual tournament, it would have been a flop. Only one team scored points consistently. But as a preseason training course, it was wildly successful. Here are just a few things that came out NAR's 3rd annual mini-game.

- freshmen had the opportunity to meet the older members and become acquainted to working with them.
- The long span of time it took place over allowed members time to ask me questions about programming as they encountered problems.
- The team had the time to work through complications with sensors and create cords before

regionals started.

- Common frustrations such as robots not running straight or sensors breaking became more familiar to younger members and more expected later down the road.

The most important part of the mini-game was that it involved everyone on the team in an experience nearly exactly the same as botball regionals and globals. NAR performed gallantly. During the last week, one member arrived every morning, forty minutes before school classes started, and worked on her robot until about five minutes before the first hour bell rang. This was in addition to the three hour meetings we had four times a week until Thanksgiving. Based on previous years, I think NAR would have lost several members during this time if not for the mini-game giving them something to work towards.

4 Leadership Choices Effecting Efficiency

Regionals started with a better team than I had expected to have. The next challenge for any team that has members that are both willing and able to do work, and have gained a little bit of experience, is organizing them so that they can work and learn efficiently, and more importantly, together. I took several steps, and probably could have taken many more, in order to ensure that we spent our time cleverly as a team.

- I started the Dropbox account that I had been thinking about earlier in the year.
- I sat down with the other senior botballer and we made a schedule for our team. We gave dates for the completion of the first build of all competition robots, the first all the way through program completion, and then set aside a large portion of time for testing, tweaking, and slightly re-engineering robots as we encountered problems.
- I made sure to have a group discussion at the beginning of each meeting with all the members so that everyone was always on the same page.

If I hadn't done these three things, Regionals would have been a mess. Imagine if during the first botball meeting in January I had just told the team to “get at it” and build those robots. One possible outcome would have been half a dozen different robots being started during the first day, followed by weeks of “Oh! You're doing that already?” and last minute programming. Another possible outcome could have been that the team members would simply have been lost. Neither would have been a good thing in terms of learning about robotics. The Dropbox account was the most important thing though. I used it primarily so that the documenters had easy access to our code and so that the programmers had easy access to the header files of the library I was developing. But in years to come, the team will be able to look back at all our programming from 2015 by logging into the NAR dropbox account.

4.1 Putting Underclassmen on the Spot

Eventually, the younger members have to put their own work into competition. The beauty of Botball is that there is no reason to not let them do this during Regionals. There are no minimum requirements for a team to meet before they go to Globals, besides money and time. During the 2015 Oklahoma Regional competition NAR ran a robot created by several freshmen during the first DE round, and if any of the younger members had built a seeding robot as well, we probably would have ran it during one of the seeding rounds. Putting younger members on the spot makes them more aware of how integral they actually are to the team. The idea is to bring the reality of their future leadership position to light early on. If the run goes badly, the young team member is forced to ask him/herself: **“If I can't build a good robot now, how can I get to a place where I *can* build one when all the older members are graduated?”**.

It is also good to remind all team members that it is alright to fail, as long as they are making an assertive effort to improve anyway. The robot made by the younger members during Regionals ended up not scoring, but the freshmen who built it are, arguebly, far more active now then they would have been otherwise. At the time of writing this paper, one of the freshmen sent me an email with suggestions for the GCER seeding strategy. We were still a week away from our first summer meeting.

4.2 How to Make Climbing Atop Other Members' Work Easier

As the summer started, one thing I did as lead programmer was print out a sheet of all the functions names in the library I'd been working on. If the library actually solves problems, then it should be used by team members, but learning a whole new library can daunting. Having a piece of paper marked up with modules, an actual physical object to carry around as one communicates with builders, this is extremely helpful for any programmer. It not only helps them learn the library through discussion, but also encourages them to communicate with the rest of the team, instead of sitting in a corner with their computer, often making mistakes because of misconceptions about the construct of a robot.

5 Conclusion, and One Final Point

After all is done, climbing is really about dedication. The tips given in this paper are supposed to be ones that when used correctly inspire members of a team to dedicate their time to the activity of robotics. But none of it will happen if no action is taken by you. My final point is about leadership. You are the leader. Forget whatever petty titles have been made up by those who are technically your superiors. You may not be the prime decision maker on whatever team you are part of, but as long as you dedicate your time and love to not only the project, but also those working on it, you are a leader of a team. In two, three, five, or even ten years, people will be striving to stand on your shoulders, often regardless of how well thought out or executed your work actually was. So it is not just a ticket to success when you work hard and think furiously about what you do on a team to make it better, but it is also a matter of respect. What kind of shoulders would you like those that come after you stand upon? What kind of legacy do you want to leave, concerning not just yourself, but the whole team? Are you, or are you not, a giant?