Efficiency in R&D
Justin Shumadine
Dead Robot Society

Efficiency in R&D

1 Introduction

Man’s pursuit of robotics is largely related to its quest for efficiency. Robots are certainly designed for a variety of reasons, but the touchstones of pretty much any autonomous system are the reduction of costs or improvement of performance, which is essentially efficiency. To sum up: humans build robots to reduce time and effort for tasks and increase performance and competence.

So if efficiency is so important to mankind and if it is such a major motivation for the field of robotics, doesn’t it make sense that the teams developing these robots also be as efficient as possible? Of course it does!

There are many philosophies about team management, documentation, communication, etc. but these are the ones which I have determined to work best for me in a broad collection of circumstances.

2 Team Management

Team organization is by far the most important administrative property of a team. The structure of a team lays the framework for planning, production, conflict resolution, and essentially everything that a team does. Thus, it is crucial that every team have some understood management structure to carry out these functions and address issues as they arise.

2.1 Democracy and Freedom

Giving all team members a voice is a crucial part of any youth program. Allowing and encouraging everyone to speak fosters creativity and open communication. This is particularly crucial for younger members, as they are the future leaders of a team. Early exposure to and participation in the decision-making process is beneficial to both the students and the team.

By consulting with the team at large about major issues, the leadership may be exposed to factors and ideas that they had not considered before. Two heads is better than one, and ten heads is better than two. A plurality of ideas and a diversity of viewpoints has proven to be generally beneficial throughout history and it has certainly served the Dead Robot Society quite well.

Perhaps the greatest benefit of democracy is the friendly atmosphere it creates. People are inherently nicer to one another when there is no power struggle. The Dead Robot Society does not have many conflicts in either design or the community, partly due to our democratic structure. With everyone able to freely voice their opinion and no one possessing the ability to outright dismiss an idea, there is little insecurity among team members.

2.2 Leadership

However, even American democracy has leadership in the form of representatives. Sometimes executive action is needed to break a stalemate or improve communication. On the Dead Robot Society, team leaders are selected by the mentorship of the team based on the mentors' observations about the students, their experiences with said students, and, inevitably, seniority. It is best to not have a freshman leading the team because no matter how competent, charismatic, or respected a person is, there is no substitute for experience.

Leaders are ultimately necessary to serve as conduits of information and executive figures. Leaders must be able to synthesize the input of their team and report to team mentors and their fellow leaders. Likewise, leaders must also communicate the directives of the mentors to the teams. Ideas, opinions, and general information flow better through established channels and in creating such channels the whole process can be greatly expedited.

On the Dead Robot Society, we have five leadership positions: two software leaders, two hardware leaders, and an overall team leader. This structure reflects the division of labor on our team, as the team consists of two specialized groups: hardware and software. By having two leaders assigned to each branch, the leaders are able to consult with each other and fill in for one another when one is absent. Having two leaders also reinforces the plurality of opinions and ideas emphasized earlier. The overall team leader is responsible for coordinating between the branches, especially in strategy and documentation. They report to the mentors on the team's progress and plans, too, as they are in communication with both the software and hardware teams and can provide a more comprehensive picture.

2.3 Mentors

They go by many names: mentors, counselors, teachers, sponsors, etc. Regardless of what they're called, mentors should serve as the gentle guides of a Botball team. They can make suggestions, present ideas and arguments, share experiences, resolve conflicts, and do any number of things, but what they should not do is directly participate in the robot's development.

The Dead Robot Society is student-driven and the vast majority of decision-making, strategizing, designing, and even instructing is carried out by students. Our mentors serve as organizers that get us a place to work, collect funds, and communicate with parents, and countless other things. We definitely would not be the team we are today without them. Mr. Gras and Mr. Newcastle have taught me, my teammates, and so many members of the Botball community so many things, but what they have not done is build is our robots.

Students must make their own mistakes just as much as they make their own successes. Mentors exist to instruct, advise, and coordinate. Based on my experiences in both Langley High School's FIRST Robotics Program and the Dead Robot Society Botball team, I can assure that a Laissez-Faire "Hands-Off" approach to instruction is by far the best.

2.4 Division of Labor

Despite all I have said about the benefits of diversity, it is not always the best policy.

Specialization is a great policy for Botball teams. Though it can certainly help to understand both sides of the construction process, it is utterly inefficient. Very few software engineers are also mechanical engineers. Just so, very few members of our robotics team have been on both hardware and software.

On Langley High School's FIRST Robotics Team, we maintained virtually no division of labor. Programmers were also wiring acuators and constructing the robot and many of the builders of the robot that did not know how to program were left without jobs. The programmers also were not the best builders, and it likely hurt the construction of the robot. Deadlines blew by, the team was in anarchy, and the whole experience culminated in a near-last place finish.

Granted, that team had problems beyond the division of labor. Proper leadership structures and communication channels can mitigate the damage of such an arrangement, but I would maintain that specialization and differentiation are still best. Hardware people should understand that there is more to programming than just plugging in numbers and functions and software people should know that there is more to hardware than nuts, bolts, and Legos. One must familiarize themselves with the parts, processes, and tricks of the trades and that comes with both time and exposure. Again, there is no substitute for experience.

Diversification has its merits and benefits, especially to an individual's human capital. It can be especially beneficial to someone in a leadership position to understand what the other branch of labor is doing. Senior members, too, may benefit from knowledge about the other branch. However, so long as both branches maintain competent workers, it really isn't an issue. Our team has never encountered an issue where something didn't work because a hardware person couldn't understand the software or vice versa. So long as the proper leadership structures are maintained and the communication channels are in place, a specialized workforce can prove immensely beneficial to the team.

2.5 Instruction

Students can be great teachers. Though our mentors have certainly taught everyone quite a bit, it is important to recognize that students can teach other students. More experienced students, in particular, can take on advanced concepts while also teaching them to newer students.

Throughout their Botball career, Dead Robots transition from grunt work to higher functions. Even those not in formal positions of leadership can take a hands-on approach to teaching. On the hardware side, we often have juniors and seniors give feedback on underclassmen designs and instruct underclassmen about the various staples of hardware like gear trains, servo mounts, claw mechanics, etc. On the software side we practice paired programming, where a new student will be paired with a more experienced one that can instruct and supervise the younger student.

These structures create a self-perpetuating student culture that can preserve the practices of the team and the knowledge acquired through years of learning and experimentation.

3 Documentation

Documentation is boring, but it’s important. It is not only impactful in this competition, but also mandated by any engineering program or office. Costs, data, observations are all required, so learn how to record them out efficiently and properly.

3.1 Who Should Do It

We at DRS usually outsource documentation to the leadership, as they tend to have the best overall view of what’s going on and they are often the most competent and senior members of the team. Though workers are often consulted for input, leadership usually oversees all documentation.

There are a number of benefits of farming documentation out to leaders. Not only are they perhaps the most qualified to do it, but it is also inherently beneficial to assign such tasks to a small group of people. Documentation is not like brainstorming. It is in fact the antithesis because it records that which has already been done. A concentrated voice can provide a uniform interpretation and maintain fluid tone and uniform structure. If several people were to write different parts of the documentation, it may appear disjointed and varying in things like detail or grammar. Even a shift in preferred pronouns or verb tense can matter. If one person were to write in the past tense and another in the present, or if one writer were to use “we” and another used “I” or another subject, then the reader could be confused as to when things occurred and who did them. Consistency is best in documentation.

3.2 Good Practices

There are also a few practices everyone should follow in documentation. First: gather data. Data is required in documentation by both KIPR and future employers; that is a guarantee. That said, make observations and record quantitative data as you test designs. There's no more reliable a way to prove what design is best or whether something will work than with cold, hard evidence.

Test early and test often. Figuring out what is feasible at the beginning of the season is crucial for developing a long-term strategy and setting appropriate goals. Early tests as simple as observing how quickly the robot can navigate the board can help determine what a team can expect to accomplish.

Test before documentation! Nothing is more annoying than preparing documentation to find that it requires a sample of data. Oftentimes this data must demonstrate how one design improved or did not improve upon another design, which requires the documenter to go back in time and rebuild old designs just to prove that they did not work. It takes more time than testing them earlier and is incredibly frustrating. Not to mention, it helps to have a quantitative basis for a team's decisions as they are making them.

Keep a log! Even creating a weekly to-do list can create a record of the evolution of a team's design and strategy. Even though the season is only several weeks, it helps to have some kind of record of what happened to trigger memory. In reflection, one may recall failed designs, tough problems, and elegant solutions; all of which are part of the documentation process!

Lastly, proofread! Even though only a few people should be writing the documentation, an outside opinion is always a good idea! Have a mentor or student that did not participate in the documentation look over the documentation. They will make sure if it seems comprehensible to an outsider and may be able to remind the documenter of experiences throughout the season. At the very least, an outside reader acts as a backup spellcheck and grammar check.

4 Communication

Establishing the proper channels of communication is not the only step a team should take to ensure that people are coordinated; using the proper tools and techniques is also essential to effective communication and coordination.

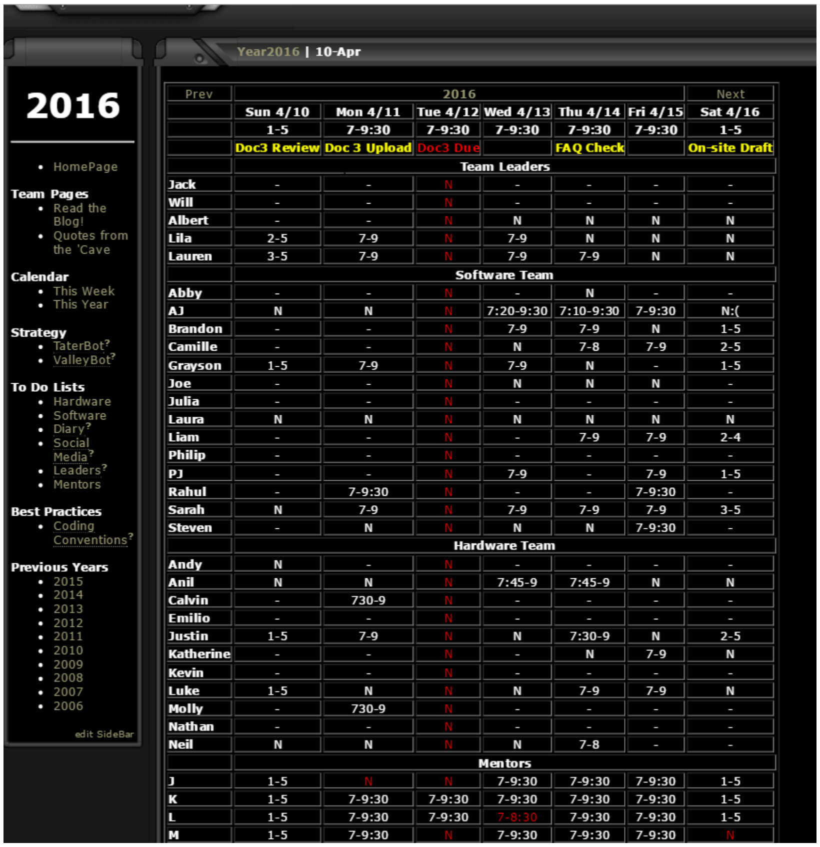
4.1 Means of Communication

The Dead Robot Society is incredibly dependent on having the proper means of communication. Unlike many other teams which are organized through a school or teacher, the Dead Robot Society is a collection of students from several schools that meet casually throughout the week at the house of one of our mentors. It is essential for us to know who can work and to spread the word of any unusual occurrences (cancellations, workshops, etc.).

For one thing, students need to know who is coming and when. It is important that both software and hardware people are present at a meeting so that they can both carry out their respective functions. Additionally, it helps to have at least one leader in attendance to record what happened and be able to communicate it to mentors and absent teammates.

It is also important to know who was where when something happened, should a change be made. Keeping a record of changes and who made them is crucial to maintaining accountability so that mistakes can avoid being repeated and successes can be imitated.

The Dead Robot Society uses many communications, varying from student group texts to parent-student emailing lists. However, our most prominent means of communication is the wiki, which can be seen in the screenshot below.



Prev	2016						Next
	Sun 4/10	Mon 4/11	Tue 4/12	Wed 4/13	Thu 4/14	Fri 4/15	Sat 4/16
	1-5	7-9:30	7-9:30	7-9:30	7-9:30	7-9:30	1-5
	Doc3 Review	Doc 3 Upload	Doc3 Due		FAQ Check		On-site Draft
	Team Leaders						
Jack	-	-	N	-	-	-	-
Will	-	-	N	-	-	-	-
Albert	-	-	N	N	N	N	N
Lila	2-5	7-9	N	7-9	N	N	N
Lauren	3-5	7-9	N	7-9	7-9	N	N
	Software Team						
Abby	-	-	N	-	N	-	-
AJ	N	N	N	7:20-9:30	7:10-9:30	7-9:30	N:(
Brandon	-	-	N	7-9	7-9	N	1-5
Camille	-	-	N	N	7-8	7-9	2-5
Grayson	1-5	7-9	N	7-9	N	-	1-5
Joe	-	-	N	N	N	N	-
Julia	-	-	N	-	-	-	-
Laura	N	N	N	N	N	N	N
Liam	-	-	N	-	7-9	7-9	2-4
Philip	-	-	N	-	-	-	-
PJ	-	-	N	7-9	-	7-9	1-5
Rahul	-	7-9:30	N	-	-	7-9:30	-
Sarah	N	7-9	N	7-9	7-9	7-9	3-5
Steven	-	N	N	N	N	7-9:30	-
	Hardware Team						
Andy	N	-	N	-	-	-	-
Anil	N	N	N	7:45-9	7:45-9	N	N
Calvin	-	730-9	N	-	-	-	-
Emilio	-	-	N	-	-	-	-
Justin	1-5	7-9	N	N	7:30-9	N	2-5
Katherine	-	-	N	-	N	7-9	N
Kevin	-	-	N	-	-	-	-
Luke	1-5	N	N	N	7-9	7-9	N
Molly	-	730-9	N	-	-	-	-
Nathan	-	-	N	-	-	-	-
Neil	N	N	N	N	7-8	-	-
	Mentors						
J	1-5	N	N	7-9:30	7-9:30	7-9:30	1-5
K	1-5	7-9:30	7-9:30	7-9:30	7-9:30	7-9:30	1-5
L	1-5	7-9:30	7-9:30	7-8:30	7-9:30	7-9:30	1-5
M	1-5	7-9:30	N	7-9:30	7-9:30	7-9:30	N

On the team wiki, students and mentors can post when they will attend meetings. Deadlines can be recorded and acted upon, too. The wiki can also report any unusual instances, like a cancellation. As one can observe on the Tuesday of the week of which the screenshot was taken, the meeting was cancelled and the team was notified so that they knew not to come.

The wiki system serves as an effective way to record attendance, as well as keep track of deadlines, strategies, and plans of the team. Such communications could be maintained through similar applications like Outlook and Facebook, but the important thing is that there is a system for recording attendance and making announcements.

Direct communications, like texting, email, phone numbers, etc. are also crucial for smaller group communications and less passive announcements. If it's urgent: use something that will send their phone a notification.

4.2 Language

It is also important that teams communicate well in person. It may seem obvious that people shouldn't yell or scream at each other, but certain practices can help a team communicate.

First: Appropriate tone of voice and language. The Botcave (The DRS home base) is a very informal environment and is no stranger to some less-than proper language. However, it should never be directed at other students, especially in a derogatory way. Just as well, there should not be yelling. If people need to yell over each other, then something is wrong with the environment or the conversation.

Second: Qualify statements and express your intentions. If something is criticism, explicitly state so. If something is advice, state so. If there is a dispute, nuance statements to clarify that you understand the other side or at least what you understand it to be. This will prevent circular conversations and mitigate frustration among students.

Third: Have a backup plan. If things get too intense, Dead Robots say "Taquitos." The safe word will immediately calm the conversation as everyone understands what warrants its use. Though it may sound crazy and perhaps something a kindergarten teacher would do, it does work.

5 Conclusion

By incorporating some of these tips into your team, you should be able to increase the overall efficiency of your operation. Certainly not all of these devices work best for your team's circumstances and some of them are a stretch for even my own. However, by mitigating conflict, catalyzing communication, and using the most efficient planning and development practices, any team should be able to create better designs faster and better sustain itself through the years.