This example would score 75 points out of 100.

Period 3: Lessons Learned Example #1
Team 16-0000

Experience Gained
We have learned a lot about the design process and how important testing is if one wants a reliable robot. We have also learned a lot about how programming works and how there can be many ways to go about one coding task. Most importantly we learned that communication is key and that everyone must be on the same page in the plan. We weren't, and the effects are now being felt.

Documentation Process
The documentation process helped us get organized and keep on task. It also helped us break our goals down into manageable sections, preventing us from being as overwhelmed.

Surprises
We were all surprised at how many things an autonomous robot can actually manage to do using the right sensors and mechanical systems. We also did not expect some of the seemingly easy building tasks to take nearly as long as they did. Just building a simple claw up to meet our preconditions took us three weeks of building, testing and redesigning.

> Marci Corey 9/15/2010 3:48 PM
>
> **Comment [1]:** The "Advice for Future Teams" section was completely omitted. Minus 5 for the missing header and minus 20 for the missing description.

This example would score 75 points out of 100.

This example would score 80 points out of 100.

Period 3: Lessons Learned Example #2
Team 16-0000

Section 1
1. Problem solving and persistence.
2. How to think creatively and to prototype things that may not work to find the solution to a problem (hardware).
3. How to look at and do things in many different ways, as opposed to doing something considered "normal".

Section 2
1. The (code) documentation process has pushed me to write cleaner, more easily understandable code with meaningful names for methods and variables.
2. Helped us understand how to write about and describe our creation to someone else.
3. Helped keep track of what had been done and what must be done.
4. Helped me be more organized and thorough.

Section 3
1. The good-natured-ness that the teams at competition have, despite the competition.
2. That everyone is willing to help you whether they are your own mentors or a different team.
3. How much fun it is!
4. How important the documentation is.

Section 4
1. Make sure that you test your code thoroughly before moving on.
2. Have Fun! It may be a competition but don't let that take the fun out of it
3. Yea, have fun; also, when making a chassis or some sort of standing square structure, remember (if you do not support it): square -> rhombus -> parallelogram -> line...
4. Do everything as soon as possible. The longer you hold things off, the less time you'll have for testing.

This example would score 80 points out of 100.

This example would receive a perfect score of 100 points out of 100.

Period 3: Lessons Learned Example #1
Team 13-0000

Experience Gained

We have learned a lot about the design process and how important testing is if one wants a reliable robot. We have also learned a lot about how programming works and how there can be many ways to go about one coding task. Most importantly we learned that communication is key and that everyone must be on the same page in the plan. We weren't, and the effects are now being felt.

Documentation Process

The documentation process helped us get organized and keep on task. It also helped us break our goals down into manageable sections, preventing us from being as overwhelmed.

Surprises

We were all surprised at how many things an autonomous robot can actually manage to do using the right sensors and mechanical systems. We also did not expect some of the seemingly easy building tasks to take nearly as long as they did. Just building a simple claw up to meet our preconditions took us three weeks of building, testing and redesigning.

Advice for Future Teams

OVERPLAN!!! Spend as much time as you need to get the perfect DETAILED plan. Then give a copy to every member of your team and continually check progress against the original plan. If we had done a better job of this our team would have worked together better and much less time would have been wasted.

This example would receive a perfect score of 100 points out of 100.

This example would receive a perfect score of 100 points out of 100.

Period 3: Lessons Learned Example #3
16-0000

Experience Gained
- Mark - I learned how to program with state programming and functions. I learned how to use loops, "ifs" and many other C commands. I also learned how gearing down loses speed and gains accuracy and how gearing up gains speed and loses accuracy. I learned how to use sensors, servos, and motors.
- Bob: - I gained more experience in CADing in SolidWorks.
- Lenny: - I learned that building a tripod helps aiming tethered projectiles. They help the aiming a lot.
- Paul: - I learned how to program the robot in an abstract way so it is easy to update your code using libraries and path planning that is based off of a real coordinate system.
- Keith: - I learned better engineering and how to cooperate with VEX and LEGOs.
- Sean: - I learned how to make libraries.

Documentation Process
- Mark: - Through documentation, I learned how to make a project plan. I learned how useful a project plan can be for keeping the team on track and planning ahead for what to work on next. I also learned that by running simulations and designing prototypes, the robot became more accurate and reliable. Designing a flow chart made the programming easier for it planned out the architecture of the code.
- Bob: - I learned that documentation can be very useful if you forget why you did things.
- Lenny: - Through documentation, I learned how to take good photos of our team robots' mechanical designs and other important key features.
- Paul: - I learned that it is lots of fun!
- Keith: - The documentation process helped me to understand that it could help us stay organized and focused.
- Sean: - By doing the project plan, I was better able to understand what was going on.

Surprises
- Mark: - I was surprised at how useful the documentation was in the long-term! I was also surprised how the programming in C is hard compared to graphical languages.
- Bob: - The season was very short this year, and it surprised me.
- Lenny: - I was surprised how the building with VEX parts is hard compared to building in LEGOs.
- Paul: - I have been doing Botball for a number of years now so not that many things surprised me. However I was surprised that edge tracking while simple in concept is difficult to implement.

This example would receive a perfect score of 100 points out of 100.

This example would receive a perfect score of 100 points out of 100.

- Keith: - The season was very short this year, and it surprised me.
- Sean: - I was surprised at how hard it is to be a team captain and how hard it was to keep the team focused.

Advice for Future Teams
- Mark: - My advice to future Botball teams is to work hard on documentation for it is rewarding in the long run.
- Bob: - My advice to future Botball teams is that CAD models take longer than you might expect and that coding does too.
- Lenny: - My advice to future Botball teams is to use rubber bands for certain mechanisms for they are faster than motors and servos.
- Paul: - "Simplicity on the other side of complexity".
- Sean: - Come up with elegant mechanisms that do the job effectively.
- Keith: - Learn how to program in C and discover how all the different motors work before the season starts.

This example would receive a perfect score of 100 points out of 100.