Using the Engineering Design Process for Design of a Competition Robot

By: John V-Neun

Introduction

This paper is designed to provide a basic understanding of the engineering design process and how to apply it to design of a competition robot. Though the engineering design process can be implemented in a variety of ways, this paper will highlight one particular method. This paper is targeted primarily at beginners but may also be useful for more advanced designers.

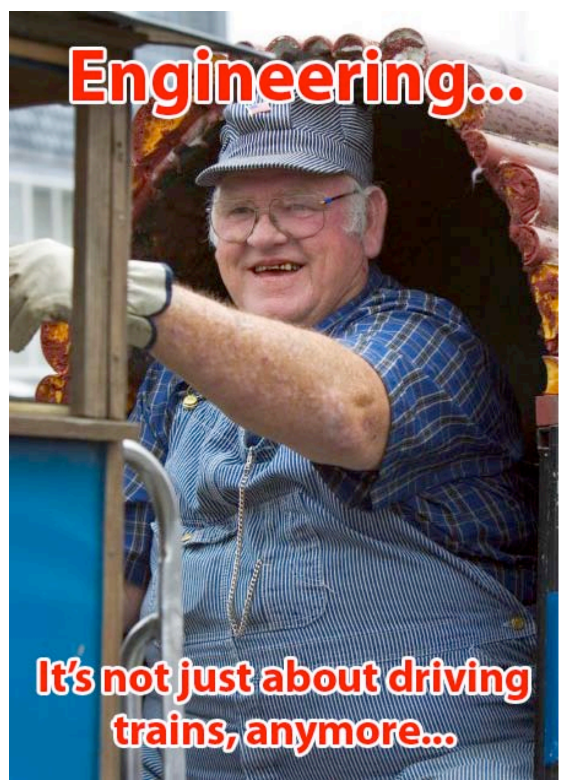
About the Author

John Vielkind-Neun has been involved in competition robotics since he was in 11th Grade at Shenendehowa High School in upstate New York. He attended Clarkson University and graduated with a BSME in 2005. Out of college John accepted a job working for Innovation First International as a Mechanical Engineer and relocated to Greenville, TX. He is actively involved as a mentor and lead engineer for Greenville Independent School District Robotics – the "Robowranglers" Team # 148.

These background details are important to note, as the author's personal experiences helped him develop his "take" on Engineering & Design, and ultimately are the source of his Engineering Design Process which is presented below. An engineer's background and experiences greatly determine their "approach to engineering". It is also important to note that the author's father is a Mechanical Engineer; so much of this could rightfully be blamed on him.

What is "Engineering?"

To understand and utilize the engineering design process, a designer must first understand what it means to be an engineer. Engineering can be defined as *following a methodical process using available resources and experience to solve complex problems*. In short, **engineering IS *problem solving***. One of the key phrases in the above definition of engineering is "methodical process." Engineers use something called the engineering design process to help them methodically find creative solutions to complex problems.



What is Design?

The next important concept to understand is what it means to design. Design is defined in the dictionary as follows:

- To conceive of fashion in the mind; invent
- To formulate a plan for; devise
- To plan out in a systematic, usually graphic form
- To create or contrive for a particular purpose or effect
- To create or execute in an artistic or highly skilled manner

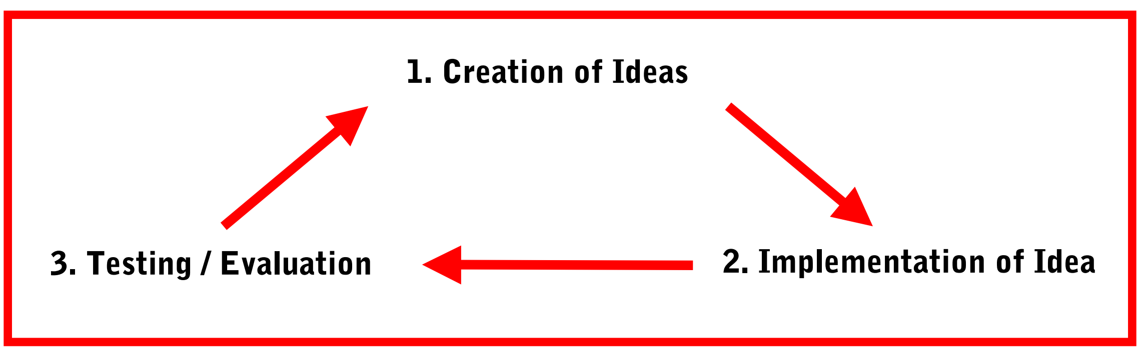
What does this mean? A simpler definition might be*: Design is thinking of and creating something new, or adapting something old to solve a problem and/or satisfy a need*. This definition has the key words "problem solving" again.

What is the Engineering Design Process?

The engineering design process is a series of steps that engineers follow when trying to solve a problem and *design* a solution for something; it is a methodical approach to problem solving. This is similar to the "Scientific Method." There is no single universally accepted design process. It seems as though every engineer has their own twist for how the process works. The process usually starts with a problem and ends with a solution, but the middle steps vary.

Think of the engineering design process as a recipe for banana bread; it can be made a lot of different ways but it's usually going to start with bananas and it's going to end with a loaf of bread. (Well, not when the author makes it, but that is a different story…) One such "recipe" for the engineering design process will be outlined below; there is no *right* design process, this is just one example.

The design process in its simplest terms can be seen as a 3-step loop:



In this simple design loop an idea is generated. This idea is implemented. After the idea is implemented, the design group would test the product or evaluate the result of the implementation. Typically, during this testing and evaluation, additional ideas are generated, and the process starts over again. This cycle and repetition is why it is said that "design is an iterative process."

Obviously this process could go on forever (or until the design group stops thinking of new ideas and stops finding problems with the design). There is a saying sometimes used that "At some point in every design process someone needs to shoot the engineer and just build the thing!"

Notes on Design Groups

Most designs are not completed by individuals, but by groups of designers working together. Not only does involving more people help spread the workload, but additional people can bring additional viewpoints to help solve a problem. As the saying goes, "two heads are better than one."

"Remember that a lone amateur built the Ark. A large group of professionals built the Titanic."

Ideally these design groups are composed of participants from a number of disciplines (i.e. Mechanical Engineer, Electrical Engineer, Industrial Design, Marketing, etc). These participants will each bring their unique skills, and unique backgrounds to the group and build on the strengths of others. A well-functioning design group will be greater than the sum of its parts.

However there can be also downfalls to working as a design group as well. Often the members of a design group will give into something called "group think." This occurs when *individual* members stop seeking a solution that excites them and start seeking a solution that is acceptable to the entire group. The quest for consensus can be dangerous. Another dangerous example of group think is overestimating one's capabilities. If the group members are too busy patting each other on the back about how awesome their plan is and become blind to the fact that they can't actually execute their plan, they've failed!

Some thoughts for a designer working as part of a design group:

- Keep an open mind. It is important to allow crazy ideas to develop. The best time for innovation to occur is early in the design process.
- Don't become overly attached to any single idea - especially one of your own.
- Do not become defensive; do not blind yourself to logic and the arguments of others. Defend your opinions and your ideas but always focus on the ultimate goal of providing the best solution possible.
- Try to stay positive, even when pointing out negatives.
- Engineering is based in logic. Do not allow emotion to interfere with the process.
- Don't let feelings be hurt if someone disagrees with you, even if they give into emotion and are (overly) harsh.
- An unjustified opinion is a worthless one. Describe WHY you like or dislike something.
- *This is NOT rhetoric, it is engineering.* It is not the one who can speak the best but the one who can provide quantitative proof that will win an argument and prove their idea is better!

"Quantitative" Arguments

The term "quantitative" is used a lot when discussing engineering arguments or justifications, but what does this really mean? Turning to the dictionary definition:

quan·ti·ta·tive (adj.)

- Expressed or expressible as a quantity.
- Of, relating to, or susceptible of measurement.
- Of, or relating, to number or quantity.

Quantitative arguments are simply ones that can be measured! In a design discussion these are extremely valuable. It is important to be as quantitative as possible at all times.

Design Documentation

When solving a problem, almost everyone follows the subconscious process. Every time they are asked to make a decision, they do this without even realizing it. This design process can be accomplished with varying degrees of formality, ranging from the subconscious process everyone does in their head to the highly documented process used in corporate engineering.

Design Notebooks

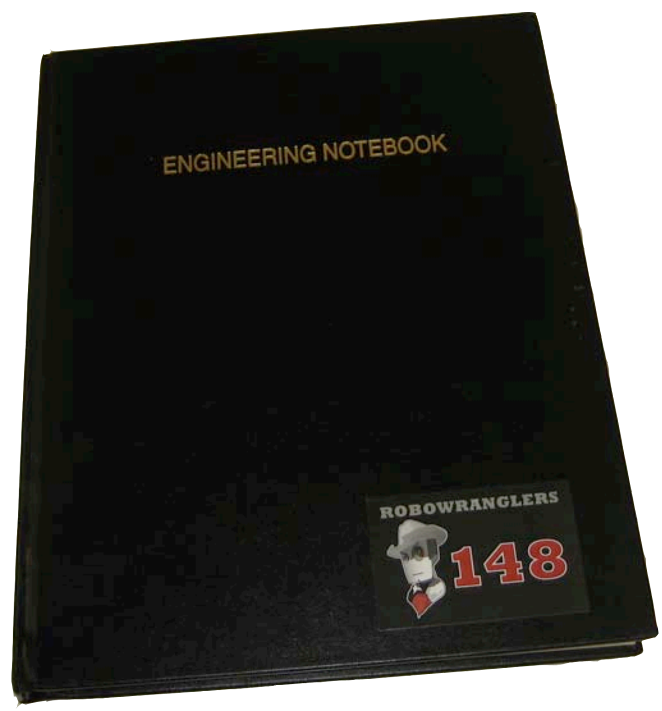
A Design Notebook is a record of the design process; it is basically a "diary" that designers keep as they progress through the process.

These Design Notebooks can take on many different formats, but they should detail each of the steps of the process combining a narrative of the progress, concept sketches, engineering calculations, pictures of prototypes, test procedures, and more. Some of the most important things to record are the decisions made, and the reasoning behind these decisions. Later on in the design process, if a designer runs into a problem and does not remember why something was done a certain way, the notebook will provide a good reference. A Design Notebook should serve as a roadmap such that any outsider can follow the designer's process, understand the choices made by the designer, and end up with the same result.

If a designer gets "hit by a bus" in the middle of a project, someone else should be able to read their Design Notebook and pickup right where they left off.

Every notebook entry should be dated and signed by the designer to provide proof of when the documented work was done; this comes in handy during any patent or intellectual property debates that occur over the design (obviously this typically doesn't apply to the work done by competition robotics teams).

Designers must determine what degree of detail and documentation is needed for their specific process. Many designers are tempted to do everything in their head, thinking that documentation will only slow them down. In truth, a more formalized process will produce a better result. Formalization will promote thoroughness; additional documentation will help avoid mistakes.



In competition robotics it is useful to keep documentation of the design, though the extent of this documentation is sometimes limited by the time available. However, as described above a documented process is a more methodical process. The notes can also be useful when explaining the design to competition judges and they will serve as good documentation for future team members who want to understand the process used.

The Engineering Design Process – Example Approach:

Step 1 – Define the Problem

This is the single most important step in the design process. Without *fully* understanding the problem how can an engineer solve it successfully? This step is often done incorrectly or incompletely and results in a failure of the design. It is important to define the *true problem,* not just the *symptoms* of the problem or the *perceived problem*.

Remember the "elevator riddle:"

There is a story about a skyscraper, which is retold to young engineers as a way of emphasizing the importance of this step in the design process. The story goes that there was a skyscraper in a major city (details will vary depending on the telling) and that the occupants of the building were complaining that the elevators were too slow. The owners of this building wanted to fix this, so they put out a call to several local engineering firms asking them for proposals.

- One firm put in a bid to renovate the office and add two additional elevators, they speculated that adding *more* elevators would cut down on elevator stops and decrease the average ride time. They estimated this would cost some ludicrous amount of money (again details vary based on the telling).
- Another engineering firm suggested renovating the building and adding some brand new, state of the art, high-speed elevators. These faster elevators would also reduce ride time. This suggestion didn't cost as much as the first proposal, but was still a ridiculous amount of money.
- A third engineering firm came back with a proposal to upgrade the elevator software. They claimed that they had constructed a brand new algorithm, which would more effectively utilize the elevators already in place to cut down on average ride time. This proposal was still somewhat expensive, but *much* cheaper than the other two.

The owners of the building were just about to hire the third firm when a fourth proposal was presented. After detailed review, the fourth proposal was immediately implemented. The fourth engineering firm suggested that full-length mirrors be installed in every elevator. When the building residents were in front of a mirror they fidgeted and adjusted their ties and didn't notice the length of the elevator ride. This proposal didn't cost the owners very much at all and was dubbed a great success. The fourth company understood that the real problem wasn't that the elevators were too slow, but that the residents *thought* the elevators were too slow…

In competitive robotics there are typically numerous problems that need to be solved by the design team. The further designers get in their robot design, the more problems come up (the main problem is often broken down into smaller problems). Early in the robot design the problems may be more "big picture" and later they will become more "detail oriented."

Some sample problems a designer may encounter that need to be solved, and questions that need to be answered can be seen below:

- What is the most effective strategy for playing the game? How do we win matches?
- How can the robot score the most points during the match? How do we score more than our opponents?
- How fast does the robot need to move?
- How can the robot pick up the game object?
  - How can the robot pick it up quickly?
- How many game objects does the robot need to hold?

These problems and questions all have many answers; some answers are better than others. How does a designer go about finding the "correct" solution or the "correct" answer? That is where the rest of the process comes into play, but until the correct problem is defined it can never be solved!

Step 2 – Generate Specifications

What are specifications? A specification is defined as an explicit set of requirements to be satisfied by a material, product, or service. In this case, specifications are requirements for the solution of the problem defined in Step 1 above.

Specifications typically come from two places:

1. Design Constraints
2. Functional Requirements

What are constraints? A constraint can be defined as a condition that a solution to a problem *must* satisfy. Constraints, in short, are restrictions. What are functional requirements? Functional requirements describe how well the finished solution must perform.

Specifications outline WHAT the solution will do and how WELL it will do it, not HOW it will do it. In competitive robotics the specs would describe WHAT the robot does, not HOW it does it. Thinking too much about "how" at this stage in the process can be counter-productive and may stifle creativity. At the same time, designers needs to keep the "how" in the back of their minds because they need to have a basic understanding of what is possible. (Specifying that a new blender will run continuously for 1 year off a single AA battery is not reasonable.)

In competitive robotics, designers are presented with some challenge or game in which their robot will compete. This challenge often includes a manual containing a series of restrictions and requirements that every robot must fulfill; these are design constraints. This is the first type of specifications a designer encounters during the process. Some examples of this type of spec are:

- Maximum robot weight
- Maximum robot size
- Allowed Control System
- Allowed Motors
- Required Battery
- Allowed Build Materials

Some specifications are also due to the resources available to the designer. Since the first set listed above are present in the competition rules they are apparent to all designers. This second set of restrictions is not always as obvious but it is equally as important to consider during the design process. Some of these may be "self-imposed" design constraint type specifications. A few examples are listed below:

- Must be manufactured without "complex machinery"
- Must fall within designer's budget
- Must utilize parts the designer already has

Understanding Your Limitations – "Know Thyself"

One of the most important parts of successfully generating Design Constraints in Competition Robotics is to understand your team's limitations. Many teams are tempted by exciting designs to overstretch their capabilities (eyes bigger than their stomachs).

Every team needs to understand exactly what they're capable of so they don't end up missing a target; capabilities often depend on manpower, resources, budget, experience, and more. It is important to focus on the "big picture" when determining whether a design is achievable. When divided up each piece may seem doable, while the overall system is "too much".

Teams will often be more successful by choosing a simple design within their capabilities and executing it very well than by choosing a complex design that they are not capable of executing!

The next group of specifications comes from the designer's functional requirements for the robot. These are things the designer believes the robot should be capable of and are performance based. Many of these are related to the challenge placed before the designer:

- Robot can run for 2 minutes continuously without draining its power supply
- Can hold 10 game objects
- Can be controlled by a student driver from 40 feet away who is looking through blurry glass
- Can push with 150 lbs of linear force
- Can accelerate to top speed in less than 1 second
- Can drive at a top speed of 10 ft/sec

It may be difficult or even impossible to generate this third type of spec early in the design process, as most of them are dependant on the nature of the design and how it progresses. These are more common during some of the sub-design processes than for the overall system.

Step 3 – Specification Ranking

All specs are not created equal, some are more important to the design than others. Designers need to think about what is most important, and why. Specifications are often ranked in some way to denote their importance. One such scale is:

- **W = Wish** (not that important, but it would be nice if it is possible)
- **P = Preferred** (important, but the project won't fail without it)
- **D = Demand** (critical to the project, MUST be included)

With these, a designer would go through and rank the specifications. These provide a good "check" for the designer at the end of the project. It is easy to go back down the list of specifications and see how well the design fulfilled them.

Designers must make decisions about what is most important when they apply these rankings. Ranking the specifications in this way will also make it clearer in the designer's mind what to focus on. Some rankings are easier than others, for instance the constraints REQUIRED by the design challenge itself are obviously ranked as "Demand."

When creating specifications some designers will list several similar specifications at different rankings to show varying degrees of importance. An example of this can be seen below:

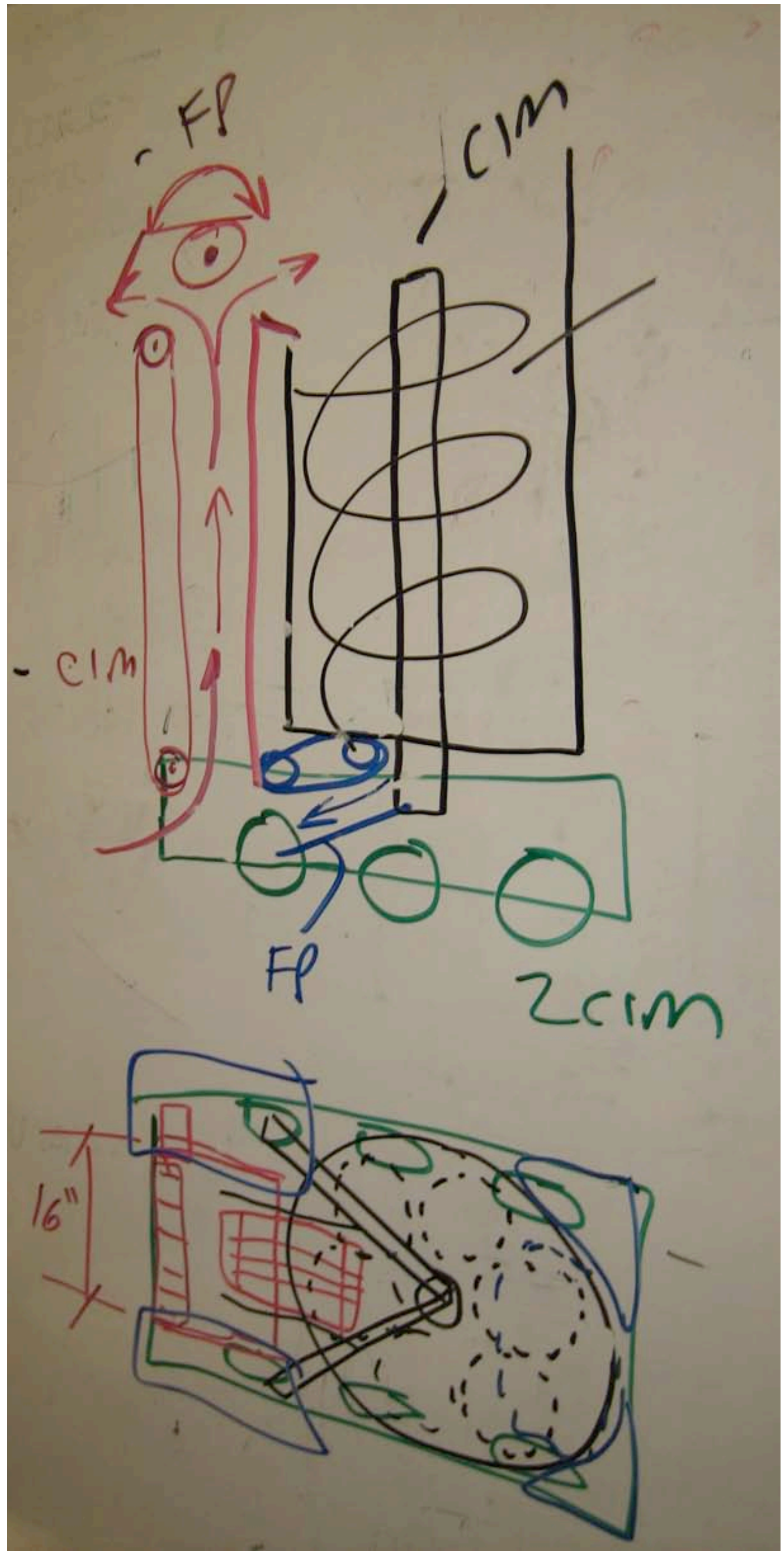
- Robot can hold 5 game objects – Demand.
- Robot can hold 10 game objects – Preferred.
- Robot can hold 15 game objects – Wish.

In the above example the specifications make it clear that the robot MUST hold 5 game objects, if possible it should hold 10, and the designer would be very, very happy if it held 15 game objects. Through the use of good specifications and ranking, it is possible to outline exactly what requirements the design team should follow and what goals the design team should strive to meet.

Step 4 – Generate Concepts & Alternatives

Everyone does the same thing when faced with a problem or a decision to make, they think of alternative courses of action; even if they do this subconsciously. Formally documenting this intuitive action may help when solving complex engineering problems. This step in the design process involves **figuring out "HOW" to accomplish the "WHAT" from the specifications**. This is not an easy process. Designs do not need to be fully realized, they should be rough concepts. Two words: *Napkin Sketches*.

This is a step that requires some creativity. Some of the most commonly asked questions are "How did you come up with that?" and "Where do you get your ideas?" Ideas come from everywhere! Inspiration can come from anywhere!



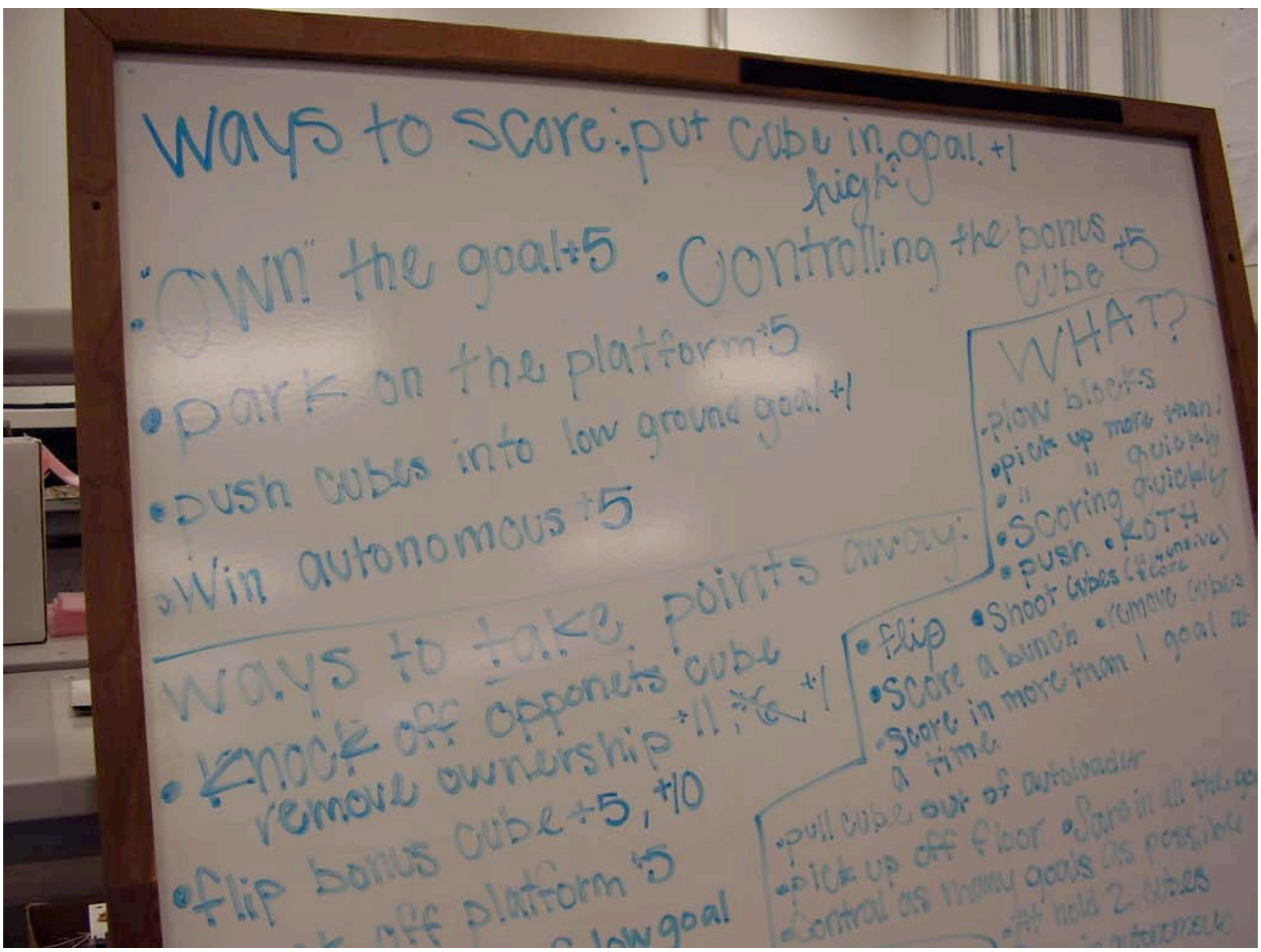
The keywords here are: "imagination" and "think." This is where the designer needs to brainstorm multiple ways to fulfill the specifications. It is important to remember to look for inspiration everywhere. A common mantra is, "*steal from the best, then invent the rest.*" Good designers will look in the world around them and try to find solutions to adapt to their problem and build off. Innovation is also important early in the design process (don't wait to innovate, always put innovation first); there is a good balance to be found between "thinking outside the box" and "using pre-made designs."

Often combining two ideas or compromising between two different suggestions may yield a good concept. Again, improvements and innovations early in the process will yield better results later in the process.

It is important not to settle for mediocre concepts and to strive to find the "right" solution. Often this "right" solution reveals itself. Designers will often comment that "it just feels right." The "right" solution will just seem elegant. Unfortunately it is not always this easy, and elegance is not always so apparent.

In competition robotics there are a number of concepts that need to be generated. Teams need to generate concept strategies, concepts for the overall system, and concepts for individual subsystems and mechanisms. Some of these systems will be dependant on and influence each other. The team's strategy will affect the overall system design, which in turn affects the different subsystems, but each of the subsystems will also affect the overall system.

These concepts are typically generated in brainstorming sessions involving the whole competition team. Concepts are recorded as diagrams, sketches, and descriptions. Many people have heard CAD described as "Computer Aided Design," but don't underestimate the importance of "Cardboard Aided Design," draw ideas on pieces of cardboard. (Everyone knows this method works much better if the cardboard used is from a pizza box.)



Brainstorming – Group Creativity Technique

One technique utilized in several steps of the engineering design process is "brainstorming." Brainstorming is a critical part of solving any problem. It is defined as a group creativity technique designed to generate a large number of ideas for the solution to a problem. The key aspect of brainstorming which makes it a successful technique, is its **focus on *quantity* of ideas generated instead of *quality* of ideas generated. *Many* ideas are generated in the hope that a *few* good ideas will develop**.

During the brainstorming process a designer or group of designers (this works much better with a group) makes a list of possible solutions to the problem being discussed. Those participating in the brainstorming are encouraged NOT to make any judgment on the ideas proposed, only to compile a large list of ideas. Brainstormers are encouraged to provide as many ideas as possible, even ideas that seem "silly." The reason for this is that even a silly idea may start a spark that inspires a great idea later in the process; creativity can come from odd places. One of the most difficult aspects of being part of a brainstorming process is to keep an open mind throughout the whole thing.

Record EVERYTHING, write down every idea developed, nothing is too wild!

Brainstorming Exercise

Sometimes it is helpful to do a group exercise to help students understand how brainstorming works. One such great exercise is to challenge the team to come up with as many alternative uses as possible for a computer mouse. This exercise works best with an old-school corded mouse with a ball. Students will be slow in suggesting uses at first, but the initial ideas will inspire other students to come up with uses of their own.

Step 5 – Prototyping

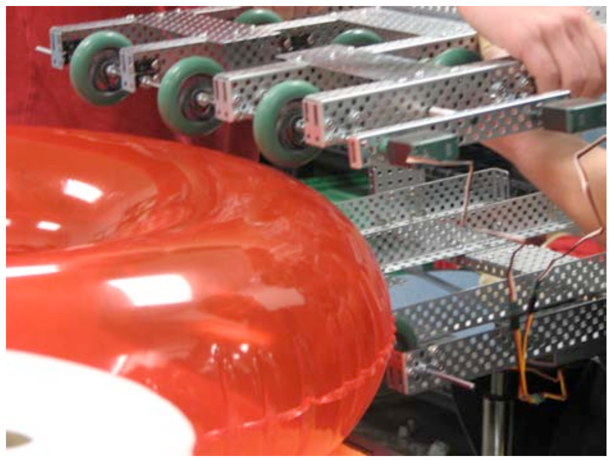
Take the Napkin Sketches and make them “real”...

This step in the process is where a designer chooses several of the concept solutions and prototypes them. The goal is to LEARN how the solution will function in “real life” and how it interacts with a real environment. This is also where a designer will start to determine which design concept will work the best. These prototypes are designed to be crude, but functional enough to be educational to the designer. The keyword here is “LEARN.”

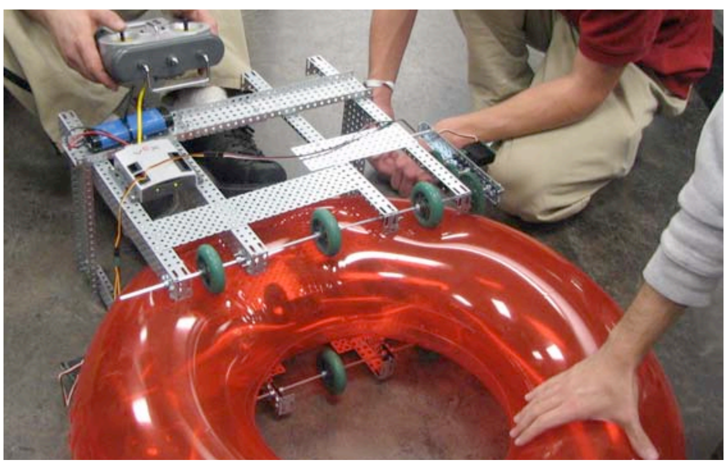
You don’t need to prototype everything, just the things you want to work!

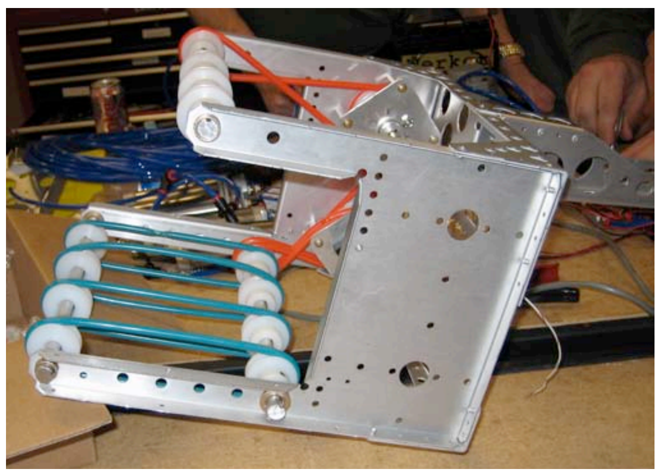
In competitive robotics, the robots must often interact with their environment and designers must learn the nature of these interactions to be successful. Test in “real world conditions” to see how things interact, find places for improvement EARLY in the design. The second stage of “Cardboard Aided Design” comes during this step in the design process; often a crude cardboard mockup is all it takes to learn a critical detail about a design (i.e. “Ohh no, it doesn’t fit in the sizing box!”). Try to see how each concept fulfills the specifications generated in Step 2. Not all prototyping needs to involve building; it is also important to do some basic testing (i.e. learn how a ball bounces, or how much force it takes to slide a goal).

Many designers will use scale models as a way of experimenting with design concepts. This is especially useful for “packaging problems” where a designer is trying to fit multiple mechanisms within a limited area. There are a number of robotic kits that are well-suited to this type of prototyping (i.e. the VEX Robotics Design System) which allow for the creation of fully functional models.



Sometimes these models can even be “full-size.” Shown below are pictures of a model claw built from VEX components and the final production claw designed from the model. This model claw was used to determine the optimal roller position; it was easy to move the rollers around until the most effective configuration was found. Once the prototype was functioning well, the designers measured the critical dimensions (roller position, tube compression, roller diameter, etc) and copied them for the final design!





Alternative Prototyping – How does one prototype "Strategy?"

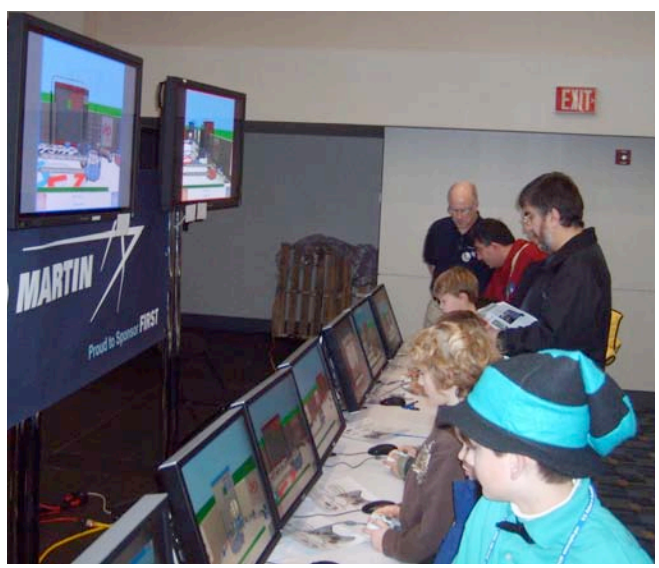
In competitive robotics, a team must not only design a robot, they must first determine the match strategy their robot will utilize. Unfortunately, it is not really possible to play matches before the competitions begin. So how do teams determine how the game will play out and figure out which match strategies will work best? There are several ways to accomplish this.

One of the most effective ways to understand how matches will play out is to figure out how long it takes to accomplish specific tasks. This will allow the team to estimate exactly what robots will accomplish during a typical match (i.e. a good robot will be able to score 3-5 tubes a match, a great robot will be able to score 6-8). Some teams base these estimates on their prior experiences, or by running time trials with their old robots.

Another way is to simulate match play. Some teams use an exercise in which they act out matches with students pretending to be robots. During these role-playing exercises, it is sometimes beneficial to assign different capabilities to each robot (i.e. this robot can hold 3 tubes at a time and score them all at once, this robot can score tubes one at a time, this robot only plays defense). By making different match scenarios with different robot strategies it is possible to see which strategies are most effective. One way to do this is using an exercise known as "STUbots" in which student team members pretend to be robots. Some have been known to use virtual simulations for this purpose.

"STUbots" courtesy of Team 116

"STUbots" courtesy of Team 116



"Match Simulator" courtesy of Blake Ross and Lockheed Martin

Step 6 – Choose a Concept

At this point in the process the designer or design group has several different potential solutions for the problem. This step is where the designers will use the lessons learned from their prototyping and determine which concept is "best" and go forward with it. This is not always an easy decision. Sometimes the "right" solution just reveals itself. Other times it is difficult to even define "best." Compare how each concept fulfills the specifications and see if one is significantly better than the others. Look for the simple and elegant solution.

"When I am working on a problem I never think about beauty. I only think about how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong." - Buckminster Fuller

"Watch Beatty's simple arm made from $10 worth of PVC out perform my finely machined mechanism with aircraft quality bearings yet again." – FRC Mentor

It may not seem like it, but these quotes are not contradicting each other. "Elegance" does not refer to "bling." Well-engineered does not mean "CNC'd." Simple elegant solutions are beautiful even if their implementation is ugly! Ugly concepts can also be implemented beautifully. but beauty may only be skin deep...

When you search for "elegance" make sure you have "elegance" properly defined!

In the event that there is no "apparent" solution, a more methodical approach must be used to make the decision. Design Groups provide their own unique challenges. Getting a group of engineers to agree on a design can be like nailing Jell-O to a wall.

Group Decision Making

When choosing concepts as a design group, it is tempting to rely on a vote to make the decision. Some people, however, believe that "VOTE" is a 4-letter word, and should be avoided at all costs. A vote is nothing but an unjustified opinion, and an unjustified opinion is worthless. When it comes to design decisions it is better to talk through things and make a logical decision by building consensus. It is important to be as quantitative as possible; don't just say something is "better," say it is "14.8% lighter" and prove why.

In some cases the decision-making is not made by the whole design group, but by a smaller leadership group or even by a single leader. In this situation the leadership is responsible for impartially comparing each of the alternatives and then choosing the course of action. This method does not always work well, especially if the rest of the design group does not recognize the authority of the leadership and questions the final decision. However, this method can be useful in preventing "stalemate" situations where no consensus can be reached. To help get the group's approval, some leaders will try to use a form of consensus-building leading up to the final decision.

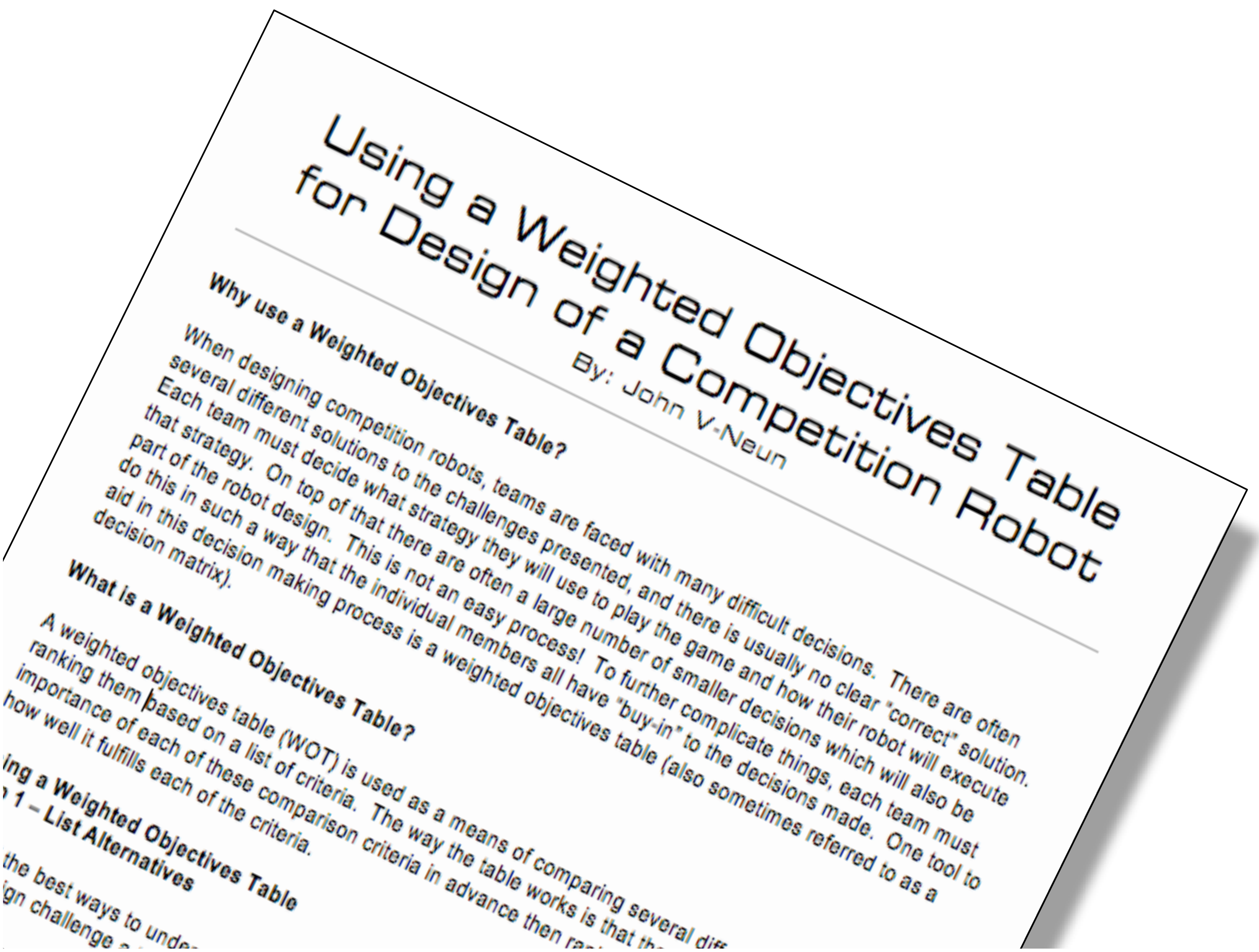
In competitive robotics, the golden rule often applies:
"*He who controls the gold, makes the rules..."* or:
"*Make sure whatever decisions you make keep your sponsors happy."*

Weighted Objectives Tables

One tool used to help during the concept selection stage of the design process is the weighted objectives table (WOT). The weighted objectives table can be used to help designers choose between several options based on several criteria. The WOT is an especially effective tool because of how it helps a designer compare alternatives based on what is “most important” to the designer.

Comparison Criteria	Weight	Roller Claw		Pinchy Claw		Scoop	
		Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Cost	5	3	15	3	15	5	25
Complexity	10	3	30	2	20	5	50
Weight	5	4	20	2	10	5	25
Tightness of Grip	5	3	15	5	25	1	5
Required Driver Precision	15	5	75	3	45	2	30
Speed of Grab	10	4	40	3	30	4	40
Total:	50		195		145		175

For more information on using a weighted objectives table in a design process, refer to the whitepaper “Using a WOT for Competition Robot Design” by John V-Neun.



Using a Weighted Objectives Table for Design of a Competition Robot

By: John V-Neun

Why use a Weighted Objectives Table?

When designing competition robots, teams are faced with many difficult decisions. There are often several different solutions to the challenges presented, and there is usually no clear "correct" solution. Each team must decide what strategy they will use to play the game and how their robot will execute that strategy. On top of that there are often a large number of smaller decisions which will also be part of the robot design. This is not an easy process! To further complicate things, each team must do this in such a way that the individual members all have "buy-in" to the decisions made. One tool to aid in this decision making process is a weighted objectives table (also sometimes referred to as a decision matrix).

What is a Weighted Objectives Table?

A weighted objectives table (WOT) is used as a means of comparing several diff ranking them based on a list of criteria. The way the table works is that th importance of each of these comparison criteria in advance then ra how well it fulfills each of the criteria.

ing a Weighted Objectives Table

1 – List Alternatives

the best ways to unde

gn challenge

Step 7 – Detailed Design

Once a final concept has been chosen, it needs to be made into something more "real." The goal at the end of this step is to have a full design that can be constructed, or a plan that can be implemented. Some of the pieces that may be generated during this step are CAD Models, Assembly Drawings, Manufacturing Plans, Bill of Materials, Maintenance Guides, User Manuals, Design Presentations, Proposals and more.

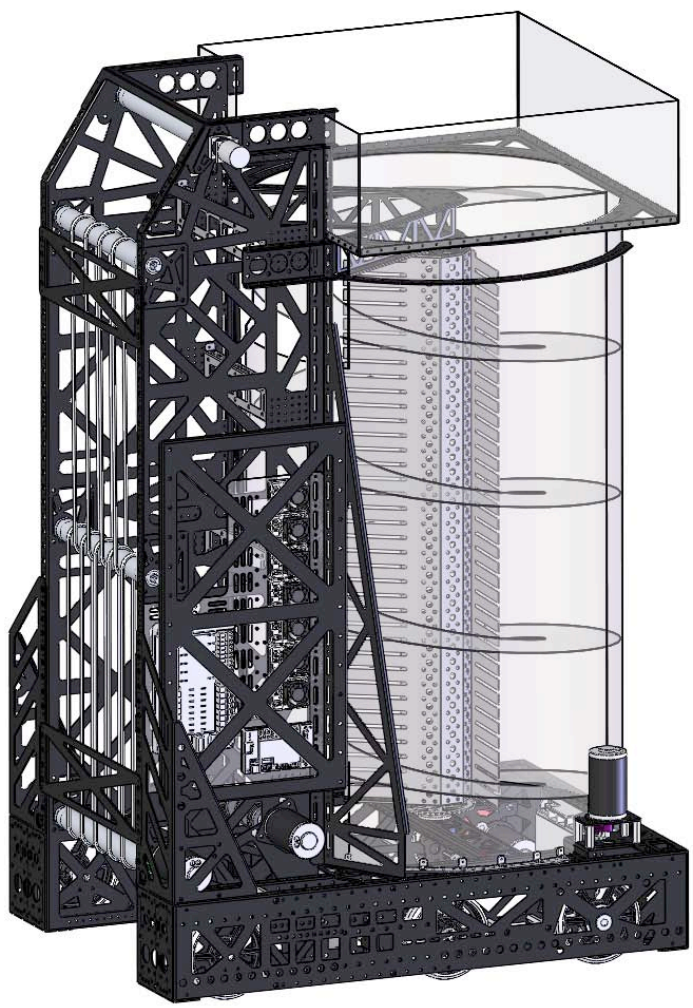
These designs will start off very crude and evolve as more details are added. It is not practical to start by detailing every piece of the solution until one sees how the pieces fit together. One tool for use during the initial design is something called "Crayola-CAD;" this is when a designer models the overall system in CAD using crude representations of its pieces. These crude pieces are then refined into more detailed pieces that are part of the final design.

This step in the process is when design calculations are completed. These calculations can refer to optimizations of gearing, material strength, weight, cost, and more. In an abbreviated design process it is not always possible to fully optimize all aspects of the design. Depending on the project, sometimes getting it "close enough" is all that is needed. Many designers can plan things using simply their prior experiences and intuition rather than calculating every detail; this kind of thinking may work fine for a high school robotics project, but probably wouldn't fly when designing pieces of a spacesuit where optimization is important. Rather than optimize each piece, just ensure they can do their job; it is okay to "over build" as long as none of the specifications are violated.

In competitive robotics it is a good idea to CAD the ENTIRE robot. This can be one of the longest stages in the design process, but the work pays off. The more work put into the design of the robot, the easier it will be to make it.

The devil is always in the details of the design.

By spending time CADing all the detail, any issues will be solved before they become serious problems; it is a LOT easier to move a hole in the CAD model than to move it in real life once a part has been made. Try to think through every screw and rivet. Plan for assembly, maintenance, and use. In addition, many advanced manufacturing techniques require detailed CAD drawings of components in order to make them (Wire EDM, Laser cut, Waterjet, CNC, etc). More and more shops are "CAD to Master" where dimensioned drawings aren't used; parts are manufactured directly off the 3D CAD models.



Step 8 – Design Presentation, Review, and Approval

The detailed design must often go through some sort of design review or approval process before it can be implemented. A design review can come in many forms. Some reviews occur as a simple conversation between two of the designers. Some reviews are done as a meeting of the Design Group where they recap and check the work that has been completed and try to find any errors. Many reviews involve presenting the detailed design to a customer, manager, or some other decision-maker for final approval.

In competitive robotics the robot designer or design group needs to present the final robot design to the rest of the team or to the team leadership for final approval. Sometimes a team will do a very formal design review meeting and invite sponsors, school administrators and community members to participate (this is a great way to increase the team's profile and drum up support).

Design Presentations are an important part of the engineering process. Many engineers believe that language arts type classes are not important to them, and because they are engineers, they have an excuse for poor spelling, bad grammar or poor communication skills. This could not be further from the truth. If an engineer has an idea, but cannot communicate it, it is worthless; if an engineer has an opinion but cannot express it, it is worthless. The ability to summarize, present, and defend ideas is an absolutely critical skill! This applies to verbal communication, written reports, slideshow presentations, engineering drawings, and other types of media.

The goal of a design review is not simply to approve the design; it is also to find any problems with the design or potential places where the design can be improved. During the design process, several alternative concepts were generated and one was chosen. There are many such choices made during the design process. Justifying these choices is one of the key parts of the design presentation. "*WHY did you do it like THAT instead of like THIS?*" The review group needs to ensure that the designers have done "due diligence" and that alternatives were investigated; they need to verify that the design is well thought out, not just the first thing that popped into someone's head.

Some questions that may come up during a design review:

- Why was it done this way?
- Did you think of doing it a different way?
- Why did you rule out other alternatives?
- Does it fulfill our requirements and specs?
- How can we make it function better?
- How can we make it weigh less?
- How can we make it faster?
- How can we make it more robust?
- How can we make it smaller?
- How can we make it simpler?
- How can we make it more efficient?
- How can we make this cheaper?
- How can we make this easier to construct?
- What other functionality would be easy to add?

Cost-Benefit Analysis

When reviewing a design it is sometimes important to perform a cost-benefit analysis. When performing this kind of analysis, a designer will look at an aspect of the design to see two things: what it costs, and how much benefit it provides. The designer will then determine whether the benefit was worth the cost of implementation.

“Cost” does not always refer to money. A feature’s cost refers to the resources that must be diverted to it; these could be time, personnel, money, space on the robot, weight, and more. It could also refer to items that must be sacrificed in order to implement the feature being analyzed. (i.e. “If we build a 2 jointed arm, we won’t have room for a ball intake on the robot.”)

Features that provide a BIG benefit at a small cost are the kind that should be added to the design (it is important to look for these at all stages of the process; a simple addition can often provide big results). High cost items should only be implemented if they provide a big benefit! These considerations are important ones, and designers need to keep them in mind.

Step 9 – Manufacturing & Implementation

Once the design has been completed and approved, it needs to be implemented. Depending on the nature of the problem being solved, the solutions to the problem could vary wildly. Depending on the type of solution, the implementation could also vary. The implementation could consist of using a new process that was designed, or it could consist of following a manufacturing plan and producing some physical object.

In competition robotics, this is the phase where we "build the thing." All the details done in Step 8 are used to create a finished, functional robot to compete with! This stage can involve purchasing components, fabricating parts, getting machine shops to produce pieces, assembly, and more - anything it takes to produce a final product.

There are a lot of different ways to build a robot. Each competition has different rules on allowable robot construction; some competitions are more open than others. In an open-ended competition, teams will need to select the best way to construct their robot and design it accordingly, be it from extrusion kits like 80/20, from welded frames, or from bent sheet-metal pieces.

This is also the stage where a team would produce marketing packets, award submissions, and other materials related to their competition, but not associated with the robot.

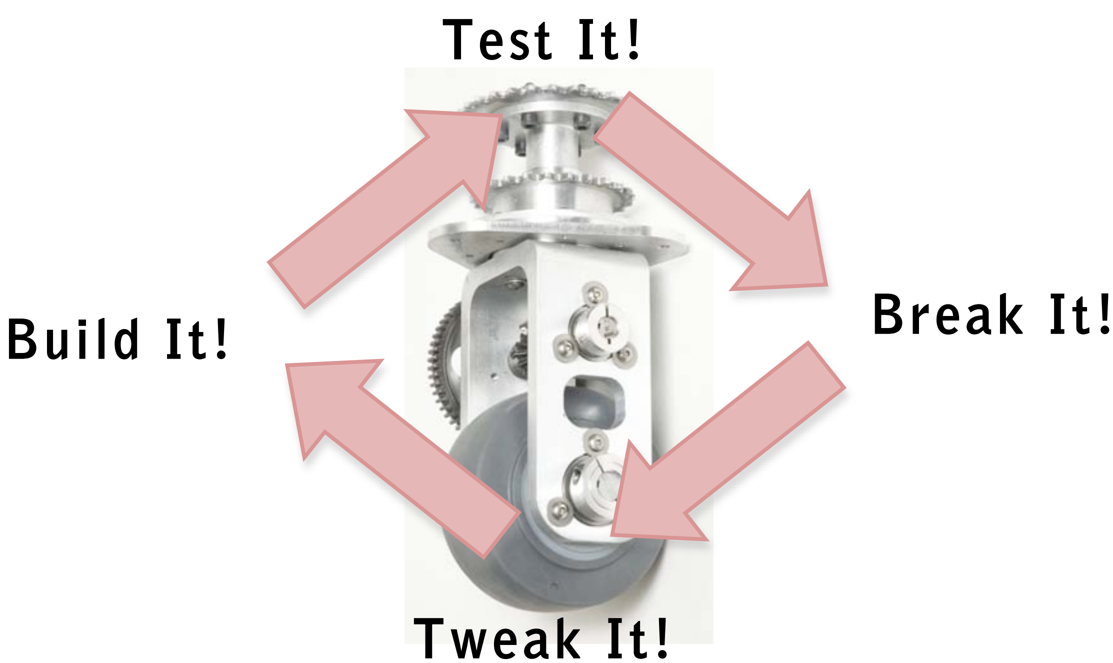
Step 10 – Testing & Analysis

Just because the solution has been implemented doesn't mean the design group's job is completed. The implementation must be reviewed to see what worked, what didn't, and what should be improved. The testing procedures and results should be well-documented. The main thing that should be determined during this stage in the process is "whether or not the final implementation performs as expected and fulfills the specifications."

So what happens if the design is not found to be acceptable? The design group must find a way to make it acceptable! The design group needs to come up with a plan of improvement to get the solution up to snuff!

Once the solution has been implemented, the analysis completed, and the design has been found acceptable, the design process is complete.

In competition robotics this testing can occur during the competition. When the robot is on the field during a match, it is apparent exactly how well it functions! However, this is not a good situation. *Most* robotics teams would prefer to know how well their robot will function BEFORE it takes the field. This is why in an ideal situation teams complete their robots with plenty of time to test and improve it. Continuous improvement is the key to success.



...eventually it comes time to "ship it."

Design is an Iterative Process! – "DONE" is a 4-letter word...

There were several mentions during the design process of repeating certain steps multiple times until an acceptable result is achieved. This act of repetition is known as "iteration." This iteration results in a better end result and is one of the most important parts of design; this is why it is said that *design is an iterative process!*

The Design Process is NOT a linear thing; it is common to jump from step to step. Design teams should NOT be afraid of going backward in the process. At any step in the process, a design team may find themselves skipping backwards to any other step. The ultimate goal is to create the best design possible by improving it over and over again. Repeat parts of the process to improve the final result.

Subsystem Design

How is a design process like an onion? They both have layers! There are often smaller design processes within the main design process. You may end up using a "mini" design process for a small part of the overall design, and then using a smaller process for one aspect of that mini process. To make this easier, the overall design is sometimes broken down into smaller chunks that can be worked on independently. These are referred to as subsystems.

There may be several parallel processes occurring at the same time, each interconnected as part of the overall system. These different layers will probably depend on each other to a certain extent, if only at some interface point. The nature of this *Systems Integration* will be discussed later on.

The greater the number of iterations a design goes through, the better the final result will be, so why would a designer every stop iterating? At first each repeat will result in large improvements to the design, but the longer the process goes on, the less problems there will be to fix and the smaller the improvements! This is known as the law of *diminishing returns*. Improvements to the design will get smaller with each successive improvement. Eventually a designer may decide that the next improvement is too small to be worth the effort, and the design is "good enough."

Some designers take longer to call a design "finished" than others because they strive for perfection. Unfortunately, in the "real world" it is not always possible to achieve perfection. Design constraints such as time and money will prevent a design team from "finishing." This is the origin of the phrase: "At some point in every design process someone needs to shoot the engineer and just build the thing!" (Some frustrated project manager who had engineers that couldn't finalize a design probably uttered this quote.)

Design & Iteration on a Schedule

While a designer might say, "*Good enough is the enemy of perfect,*" a project manager (who is nervously watching the schedule) would be more likely to counter "*Done is better than perfect!"*

In competition robotics, how do we reconcile the benefits of continuous (and potentially never-ending) improvement with the need for project completion? Simple... each team needs to set a schedule, and then stick to it. This schedule will vary greatly from team to team depending on their circumstances. If a team has six weeks to design and build their robot before they must ship it, they should come up with some sort of schedule for this time period. This schedule can vary in detail greatly. Some teams will plan out each and every step in the process while others will just do a quick overview. A sample "rough" schedule is shown below:

Sample Design Schedule – 6-week process**:**

Day 1 – Read & Learn the Game Rules, Play with Game Objects, and Analyze Scoring
Week 1 – Choose Match Strategy, Generate Concepts and Build Prototypes
Week 2 – Choose Robot Concept and Begin Detailed Design
Week 3 – Finish Detailed Design and Begin Manufacturing
Week 4 – Finish Manufacturing and Begin Testing
Week 5 & 6 – Test, Tweak, Improve, & Practice – (ITERATE!)

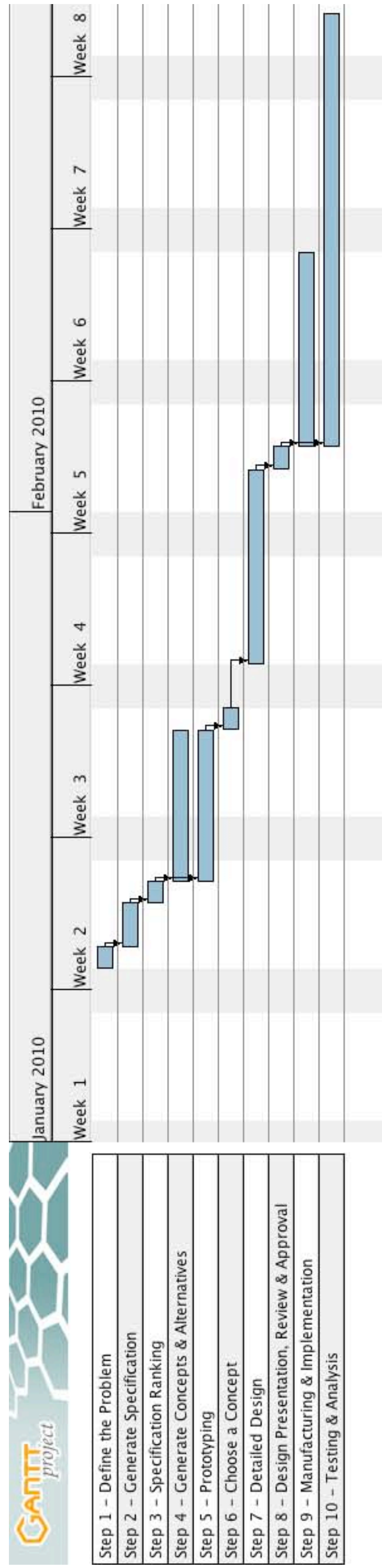
Obviously this schedule would not work for every team. Some teams may be able to rapidly manufacture their robot; this would allow them to spend more time on the other parts of the process. By having some rough outline of the schedule, it will help a team effectively choose a design they are capable of building. One shouldn't choose a design that will take 4 weeks to build, because there won't be any time for testing and improving!

In the above schedule, the most important part of the process is in week 5 & 6. Regardless of what design is chosen, if a team has two weeks to improve it and make it reliable, they will be successful. This is "planned iteration," the team knows how long they have to iterate their design, and they have a good idea of how far along in the process they *should* be. Iteration will not just happen during these final weeks, it will happen during EVERY stage in the process. This is great, as long as everyone understands there is a schedule to keep and they can't spend 4 weeks choosing a robot concept.

The schedule is not always set in stone; ultimately the only fixed dates are the project start date and the robot completion deadline (a ship date, or the date of the competition.) Everything else is likely to shift as the process unfolds. Many teams in competitive robotics know these shifts will occur, so they don't even bother trying to plan a schedule in detail.

Advanced Scheduling Tool – Gantt Charts

One way to illustrate a project schedule is using something called a Gantt chart. A Gantt chart is a bar chart that shows each piece of the project displayed on the overall project timeline as bars stretching from their start to finish. If an aspect of a project cannot begin until another has been completed, the start of the 2nd bar is linked to the end of the 1st; this is a good way to demonstrate how some pieces are dependant on others. By looking at the Gantt chart it is easy for a designer to get a snapshot of how far along the project should be, what needs to be in work, and what is behind schedule.



Systems Integration

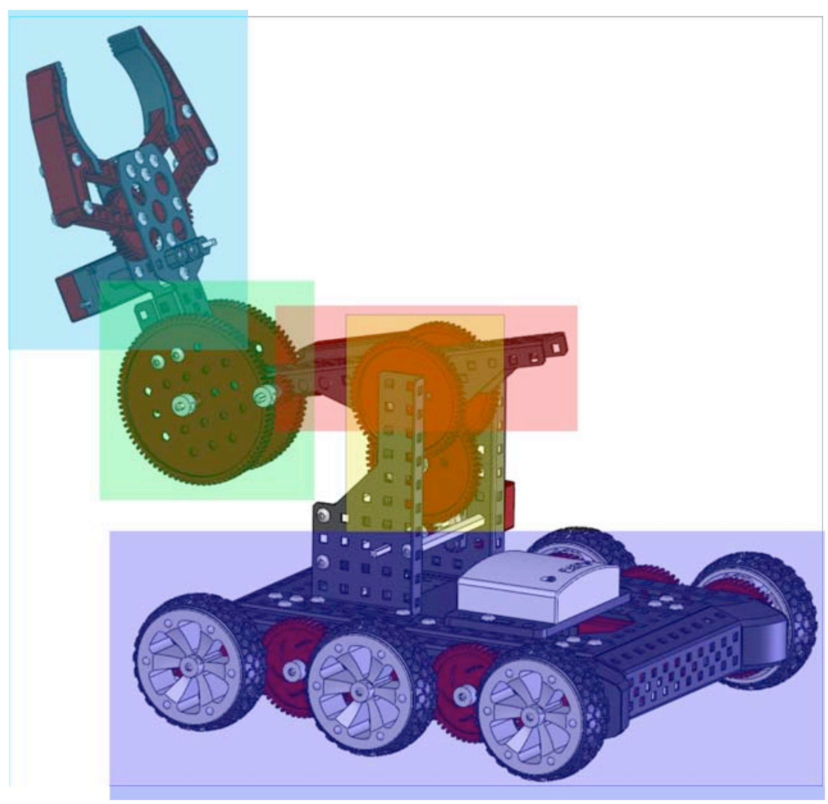
One of the most important concepts in design is that of systems integration. It was discussed earlier how a design is often divided into subsystems. Systems integration refers to the way that these individual subsystems are combined during the design process into one cohesive product. This is not something that happens at the end of the design process after each subsystem has been designed, but rather something that is integral to the process. Ideally each subsystem works well and supports the other subsystems of the robot in a way that the whole becomes greater than the sum of the parts. It is important during the design process to think about the ways each subsystem will affect and interface with each of the others.

In competitive robotics the robots are typically divided down into subsystems which perform different functions. Some of these systems will stretch throughout the robot while others will consist of only a single mechanism. An example list of subsystems is shown below:

Example Subsystems of a Competition Robot:

- Power
- Control
- Sensors
- Pneumatics
- Drivetrain
- Robot Arm
- Shoulder Joint
- Wrist Joint
- Manipulator

Each of these subsystems could be designed independently, but each is dependant on all the others. In order for the overall robot to function effectively, each of these systems must work together. In order to design any one of these systems, one must have knowledge of the others. Each of these subsystems will have its own individual design process as part of the overall design process; any requirements on the way the subsystems interact as part of the systems integration would be treated as Specifications (design constraints) in Step 2 of these sub-processes.



For example, if the robot has a maximum size requirement, each subsystem may need to “nest” together such that the overall system “fits in the box.” In this scenario each subsystem may be given a specified amount of “robot real estate” that it must stay within and specified ways they would interface with the others. (i.e. the claw must start within an imaginary 10” x 10” x 12” tall box located at the very front of the robot that is 6” off the ground, it attaches to the wrist joint at the lower, rear center of this imaginary box using a specified hole pattern.)

Implementing a Design Process

In competitive robotics it is not always easy to implement and successfully utilize a structured design process. Often the frantic, fast-paced nature of these competitions makes a designer wish to forego formality in favor of a more intuitive “all in their head” style. This approach may work in some applications, but it is not conducive to successfully working as part of a larger design *team* on the complex systems found in competition robotics.

The easiest way to implement a design process is to start rough and then refine things as the season progresses. The easiest place to use a more structured process is at the very beginning of the season. Many teams have a distinct set of steps they use for analyzing the game, and once they get past the initial few days of robot design, they will adapt their process depending on how things are going. Repeating this process each year can be beneficial as well; students and mentors on the team will become familiar and comfortable with the team’s schedule and will become more efficient at designing. In order to maximize this effect, some teams will do a “practice” run-through of a robot design and use the same design process for all the multiple robotics competitions they participate in (i.e. using the same design process for a VEX team, a BEST team, and an FRC team).

Iteration Examples

After discussing the importance of iteration, it is commonly asked: "*What are some examples of iteration that a competitive robotics team might encounter?*" Below is a list of potential iterations and "jumps" that might occur during a design process. This list may help budding designers better understand ways that improvements occur. It may also help them recognize ways they have *already* been iterating as they solve problems!

Causes of Iteration:

- During final testing, learning a piece of a subsystem doesn't work.
- During brainstorming, consulting the rulebook and determining a chosen strategy is illegal.
- While CADding the robot, finding that two subsystems interfere with each other.
- During robot assembly finding, there is no way to install a critical bolt.
- Calculating the weight of your robot design and determining it is over the limit.
- Weighing the robot for the first time and finding it is over the limit.
- Finding your robot arm can't reach high enough to score in the goal.
- Having a component break during testing.
- Determining that a motor isn't powerful enough for its designed function.
- Finding a leak in the pneumatics system.
- During the final design presentation, having a flaw pointed out.
- The shop manager complains that a part cannot be manufactured.
- During the design, it becomes apparent that the specifications cannot be met.
- The final design of one subsystem would function much better if another subsystem design gets tweaked.
- During the detailed design phase, the final prototype cannot be reproduced effectively.
- During specification ranking, additional specifications are determined.
- The electrical subsystem cannot be integrated into the final design due to space constraints.

Sometimes during the design process a team will find out that their execution is not lining up with their expectations. If this happens it is absolutely critical that the team step back and determine exactly why this is occurring. It may be necessary to change "the plan" to help meet the original goals and design requirements or it may be necessary to adjust these goals to become more realistic. This is especially important when it comes to meeting the design schedule; if things are not progressing as quickly as planned it may be necessary to adjust the goals such that the schedule can be met. In a competitive robotics environment anything less than 100% complete often rounds down to zero; if the arm is 90% done, it is usually 0% functional! It is okay to eliminate or scale back goals in order to complete some goals. This is directly related to the "know thyself" concept mentioned earlier; if a team fails to successfully gauge their abilities and plan accordingly, they may find themselves in this situation later in the process.

Example Design Process

In competition robotics there are a lot of ways to utilize the engineering design process. One such method is shown below. This is the process used by the Greenville Robotics Team # 148 for FRC and VEX. This method can be used for any number of competition robotics programs:

Kickoff Process (1 day):

- Outline the plan to the team in advance.
- Watch Game Animation – No talking allowed!
- Read Game Manual as a team, and paraphrase rules so everyone understands them.
- List any questions on game rules that the team feels are unclear.
- Explain “What is Brainstorming” to the team.
- Begin Game Analysis:
    - List all the ways a robot may score in the game.
    - List all the ways a robot may de-score in the game.
    - List all the possible defensive moves.
    - List any offensive or defensive “power moves” in this game.
- Do Rough Scoring Analysis:
    - Does the difficulty of tasks match the reward? Cost-Benefit Analysis.
    - Determine if there are any “scoring lock” wins (i.e. score 16 unanswered footballs, and guarantee a win, score all 3 goals and guarantee a win.)
- Brainstorm Match Strategy Options
    - Determine “WHAT” the robot will do, not “HOW” it will do it.
    - List any questions the team has about playing the game:
        - How hard is it to score the footballs? Can the football fit inside the goal?

Prototyping & Initial Design (1-2 weeks):

- Built Prototypes and perform tests to answer the questions developed during brainstorming.
- Start determining what robot strategy is best for winning matches.
- Start determining what system design best fulfills the strategy.
- Do research and find existing ideas that will work as part of the overall design.
- Test individual subsystem concepts (i.e. drivetrain, arm, claw, etc.)

Choose Robot Strategy & Robot System Design:

- Based on the lessons learned during prototyping, choose the best strategy.
- Based on the lessons learned during prototyping, choose the best overall system plan.

Detailed Design, Additional Prototyping (1-2 weeks)

Manufacturing & Assembly (1-2 weeks)

Testing & Design Iteration – “Keep improving until you can’t.” (2 weeks)

Note: As stated above, every team must tailor their build schedule to fit their specific needs and capabilities. If a team has access to rapid manufacturing techniques, they can afford to spend more time “thinking about what to build” instead of actually “building it”. This all goes back to the concept of “know thyself”; it is important to understand exactly what sort of schedule a team should be working toward (and what pace they should be working at) to meet required deadlines.
What works for 148 won’t work for everyone...

Conclusion

It should now be apparent how simple it can be to implement a structured design process on a competition robotics team. However, structure is not necessary to achieve many of its benefits. Simply by creatively tailoring the team's build/design schedule and ensuring team members are familiar with the basic design process will result in a subconscious shift toward a formalized process; team members will recognize what part of the process they are in, and subconsciously know what they need to do next. The single most important step a team can take is to educate its members on the iterative nature of design, and the ways iteration will result in a better final product.

This is powerful stuff that applies to other aspects of life, not just competitive robotics. Engineering is problem solving. How many problems does a person solve in an average day? How many times do people run through this process (in some form or another) in their head each day? Design is everywhere! If the goal of a competitive robotics program is to inspire students to enjoy engineering, these are the key aspects of engineering that competitive robotics programs need to emphasize. One could argue that it does NOT matter if a student knows how to use a milling machine or how to use CAD, these are side-notes compared to some of the big picture concepts. It is more important that students understand how to methodically solve problems in the same manner as an engineer and understand how to think like an engineer; this is the essence of engineering and one of the most powerful concepts students can learn.

To be really successful, all one needs to do is teach a student that solving problems is fun!

JVN's Personal Notes

Here are some "editorial" comments on designing for a competition robotics program from the author's experiences. These may not apply to every situation.

1. **One of the worst things a team can do is choose a strategy or robot design that they are not capable of executing.** Some teams believe that over-reaching their capabilities is a good way to learn, but at the same time failing to successfully implement a design can lead to a negative experience for students on the team.
    a. One common example of this is something called "FYSS" or "First-Year-Swerve-Syndrome." This refers to what happens when a team tries to build a swerve drive for the first time. Most teams have difficulty building it the first time, and as a result they spend the entire season struggling to get it working effectively. Teams should be wary about building a swerve drive for the first time during a competition season lest they suffer from FYSS.

2. A team shouldn't be afraid to ask for help! If a team doesn't have the resources necessary to accomplish something, they should seek out those resources. Seek help from other teams, or from the local community. If money is the problem, the team should focus on fundraising opportunities. If the problem is a lack of engineers, the team should recruit members from their community. If the team lacks expertise, they should spend the off-season learning about the topics they're weak in.

3. When some teams are deciding what to build they choose designs based on "cool factor" and "bling." While it is great that teams want their robots to "wow" the opposition, they should remember that this is not the engineering challenge they are competing in. The engineering problem the team is trying to solve does not involve cool factor. Make decisions based on sound engineering judgment and learn to use the engineering design process to solve the problem given as part of the competition, not some other problem. This method allows a team to make the robotics competition an even more effective example of "real world" engineering design. In the real world, one can't just make up your own criteria for success (i.e. "We're not trying to build the most effective machine, we just want to build a cool machine.") one needs to use the criteria for success that are given.

4. Many people hate it when they need to redo something. As a result of this they tend to try to justify to themselves and others why something is "good enough" and does not need to be reworked. It is important to try to emphasize to everyone that reworking is okay and expected because design is an iterative process. Ask team members to determine if their decision to settle for a "good enough" solution is a lazy decision or a smart decision.
    a. Celebrate failures as opportunities for learning and improvement!
    b. Note, this is not the same as saying sloppy work is acceptable.

5. There is no problem that hard work and creativity cannot solve. Don't be afraid to go the extra mile. The most successful designers and teams are frequently the ones who work the hardest and are constantly striving to improve their designs and their programs.
    a. The design phase of a robot isn't over until the robot is dusty on a shelf. The best teams will continue to improve for the entire season. Even in competition programs where the robot must be shipped it doesn't mean all design must stop. The secret is CONSTANT self-evaluation and CONTINUOUS improvement.

PROFESSIONALISM AT ALL TIMES!

ROBOWRANGLERS

148

WHAT DOES IT MEAN TO BE PROFESSIONAL?