

## Determining Design Targets

### House of Quality (QFD) Guide

Every design involves trade-offs. Do we increase the maximum speed but use more fuel and reduce the runtime of our system? Do we make our system more expensive or re-allocate part of cost budget from one subsystem to another, potentially causing one or more subsystems to not perform as well? Do we increase the allowable error rate but make our systems handle errors more easily? All of these decisions relate to setting *design targets* – desired values for all of the characteristics of our system that we as designers set. But how do we know and justify to others that we have determined the best design targets?

The answer stems from first understanding how your system’s performance will be evaluated, i.e. how will any solution be measured for meeting the needs of the challenge you are trying to solve.\*\*\* But the key to setting good design targets is both:

- Understanding how adjusting those design targets will influence the performance measures, and
- Understanding how adjusting one design target may also influence how easily it is to meet other design targets.

One of the best known methods for collectively viewing all of these impacts together is known as Quality Functional Deployment (QFD), also commonly known as creating a “House of Quality”. This guide will first introduce you to some of the key terminology and then introduce an example system. Then through a step by step process, this guide will use that example to show you how to create your own QFD, and more importantly help you and your team come together to make solid, defensible design trade-off decisions both early on and throughout your design process.

The Steps presented offer one way of marching through the QFD process but in practice many times several steps may be done in conjunction and often re-iterated upon well after the first QFD matrix has been completed. Hence, you are encouraged to read through the complete steps & example before you try to complete your QFD. At the same time, if after reading the steps you are still uncertain about how to use the QFD, try following the steps directly as they can guide you until you’re more confident.

**Step 0:** Talk the talk. An important part of having your work be recognized as being professional is to use the same terminology that professionals do. You’ve already heard the term “design targets” above but what qualifies something as a design target versus say a requirement or a constraint.

A “design target” often refers to the intended value for an “engineering characteristic” in your final design. Okay so what’s an “engineering characteristic”? An “engineering characteristic” in turn is some property of your system design that *you have some control over*: such as its speed, its weight, its size, its mean failure time, its material costs, its manufacturing cost, etc. These are all properties of your system that whose value will result from how you design your system.

Engineering characteristics are often thought of the “knobs” of your design that you can tweak but often adjusting one knob can influence others. For example, one engineering characteristic might be your overall size and another might be your system’s weight. You could make your system bigger but that might also make it heavier.

Some of these engineering characteristics may also map directly to a performance criteria. Engineering characteristics like material costs and manufacturing cost could “map” to or “influence” a performance criteria called “Inexpensiveness” or simply “Cost”. However, many engineering characteristics may have an influence over several performance criteria. For example, the size of your system may influence performance criteria such as cost, visual appeal, transportability, etc.

In order to help make the distinction, engineering characteristics are what you control, performance criteria are what your customer cares about. So ultimately your goal in your QFD is to know how adjusting your engineering characteristic knobs will help you achieve the highest performance score. The desired values you have chosen for your engineering characteristic knobs are referred to your design targets.

**Step 0 is complete when:** you have a basic understanding of what a performance criteria, engineering characteristic, and a design target are. If they aren’t perfectly clear yet, that’s okay. But they should be by the end of the Step 1 example.

As one last common piece of terminology, by the end of Step 19 you will also see why the QFD is also frequently referred to as the “House of Quality”.

---

**Step 1a:** (optional) In some cases it can be helpful to add a bit of a reminder as to what a performance measure is actually targeting. For example, “Robustness” could mean a lot of different things to different people. Hence, a short description column can be added to the leftmost column of the QFD, which is particularly worthwhile for larger teams.

In the example, this is done using the text that was written in parenthesis next to the performance criteria listed in Step 1.

**Step 1a is complete when:** You have added any optional short descriptions to the QFD.

---

**Step 1b:** (optional) In using the QFD internally, there may sometimes be other criteria that are not directly important to the customer but are important to you as the company making the product. These additional criteria are sometimes listed in the QFD using italics text and a few examples of common ones are listed below.

- *Manufacturing & assembly time*
- *Commonality in parts with other company products*
- *Profit margin*

If this is your first time using a QFD to explore your customer’s needs, you may want to complete the QFD without these additional criteria and then add them in later once you have used the QFD to better understand the customer’s needs.

**Step 1b is complete when:** You have added any optional internally important criteria to the QFD.

---

**Step 2:** Weight the importance of your performance criteria. Not all needs are equally important so each performance criteria should be given a relative weight listed in the column immediately to the performance criteria’s right. Typically this is represented as a percentage, i.e. in the Step 2 tab of the example, “Time to accomplish tasks” is assigned 13.51% of the total weight across all criteria.

Wait a second, “13.51”? How did we decide it should 13.51 versus 13.52 or even versus 15 for that matter? Performance criteria weighting is covered in the other guides mentioned earlier\* but to the right of the QFD columns you will notice some scratch work that we did as one quick example for how to weigh performance criteria. Here we rated each performance criteria separately on a 1-5 scale of importance in Column AN. We then summed all the performance criteria importance scores in Column AN to get the value of 37. We then divided each value in Column AN by 37 to get the percentage of the total importance score for each performance criteria as shown in Columns AO and AP.

This is a quick example, as we didn’t justify why one performance criteria should get a “5” and another a “4” (strong, objective justifications can come from analyzing customer data or the original challenge’s definition) but regardless of how you determine the relative importance (aka weight) of each performance metric it can be helpful later to represent this as a percentage.

**Step 2 is complete when:** you have assigned a relative importance percentage to each of your performance criteria.

---

**Step 3:** Rate your current solution concept, to the best of your ability, against your performance criteria. You may not have any of your solution designed yet but you should have an idea as to where your concept may be strongest or weakest. As you complete your design, begin prototyping and testing, and continue to refine your design, your estimates should become more realistic. But for now do the best job you can.

Sometimes it’s helpful to both provide an upper and lower estimate. The upper estimate is if everything works out the way you intend in your current concept, and the lower estimate should represent the minimum level that you can currently guarantee. If you do not have any data from any test of your system, your minimum may be very low, even zero (on a 1-5 scale).

But that’s perfectly okay to give yourself a zero – and it’s actually most helpful to be honest and hard on yourself. That zero alone helps to inform you where you have the most uncertainty about your system’s performance which is very valuable. Areas where you are the most uncertain or similarly show the largest difference between your low and high estimates are areas you should most likely focus your efforts on first.

It may feel natural to address the areas that you are most confident and hence possibly most comfortable with, and it’s okay to do that for a *very short* time if you need to help build up intuition about your project. However, it’s a terrible situation to be in where you have put significant time, effort, and resources into the areas you are confident in, only to later realize when exploring an area that you were less confident in, that the hard work you have already done is now invalid due to a need you discovered in exploring that uncertainty. Or at the very least, you might discover that you spent too much time/resources on the area you’re confident in and are now lacking the time/resources to handle that uncertain area appropriately.

Overall, scores assigned should be unitless normalized values (i.e. all on the same scale, like 1-5) otherwise you won’t be able to add them together later, i.e. How would you add \$1000 to 5m/s to 0.3% failure rate? Ways to establish objective metrics and normalize values for performance criteria is covered in the Decision Matrix guide.

**Step 3 is complete when:** You have given a unitless normalized performance score for your current concept. Even better yet, it’s best to give a low and high estimate.

As you perform more work on your design, you should hopefully see your low estimates increase (and not uncommonly see your high estimates become more realistic) but either way this can also be a great way to track your progress.

---

**Step 3a:** Multiply your normalized performance scores by the respective relative importance weights to get the current weighed scores for your concept. This is done in the example by multiplying Column D with Columns AE and AF and storing the results in Columns AI and AJ respectively.

Add the total for weighed normalized performance scores to get your total performance score for that Column. This summation number is listed at the bottom of that Column as shown in the example tab for Step 3a. This gives you your current estimate as to how well your system is meeting the overall challenge's needs.

**Step 3a is complete when:** you have the weighted performance scores and total weighted performance score for all of your performance score columns.

---

**Step 4:** Repeat Steps 3 and 3a for any other concepts you are considering as well as any potential competitors.

It's sometimes natural to initially think "What we're doing is so novel, there aren't any competitors out there!" However, you may want to ask yourself, "Well if that's the case, how are people now addressing the needs your system is aiming at handling?" There is likely to be some solution out there to address at least some of the needs your system is targeting. Yes, there may not have been anything quite like the first air conditioner when it was first developed. But there were other ways that people meet the need of cooling down an interior space. Before air conditioners, people used fans. Before fans, people just opened the window. Now you might initially argue that opening a window is a horrible solution but when you consider potential performance criteria such as cost, solution energy usage, installation complexity, maintenance, solution lifespan etc., "opening the window" actually has a number of good things going for it.

A valid contrary argument could potentially state though that "opening a window" however might not meet all of the basic requirements for your particular challenges needs – perhaps you need a solution that guarantees a temperature in the room to within a certain tight tolerance; let's relate this need to a performance criteria we'll call "Temperature tolerance". In this case, you might assign "opening a window" a zero "Temperature tolerance" score. However, it still might be good to include "opening a window" as a competitor since it does so well in so many other performance criteria. Plus it may help you to brainstorm other concept solutions later – or at the very least including "opening a window" in the QFD can help to formalize why "opening a window" is not a good solution.

**Step 4 is complete when:** you have added additional columns to compute the total weighed performance score for all current concepts and competitors.

The area that all of the performance scores are recorded is often referred to as the QFD "back porch" which will become more apparent as we build the QFD.

---

**Step 5:** Take a break – especially if you haven't done a performance criteria decision matrix on your current concept and competing solutions prior to starting your QFD. This is also a great time to discuss you findings with your teammates. Make sure you all agree on the performance criteria and their weights. Are there any additional competitors you should consider – in general, the more the better, as other ideas may inspire new ideas opportunities, or help you recognize

potential weaknesses in your current solution concept. Recognizing weaknesses is a great thing! – especially when you’re still early in your design project and have the time & resources still to make your project stronger. How else are you going to truly get better, if you don’t recognized where you need to improve the most?

**Step 5 is complete when:** You are at least a bit more refreshed and ready to look at things from different perspectives with regards to your own current solution concept.

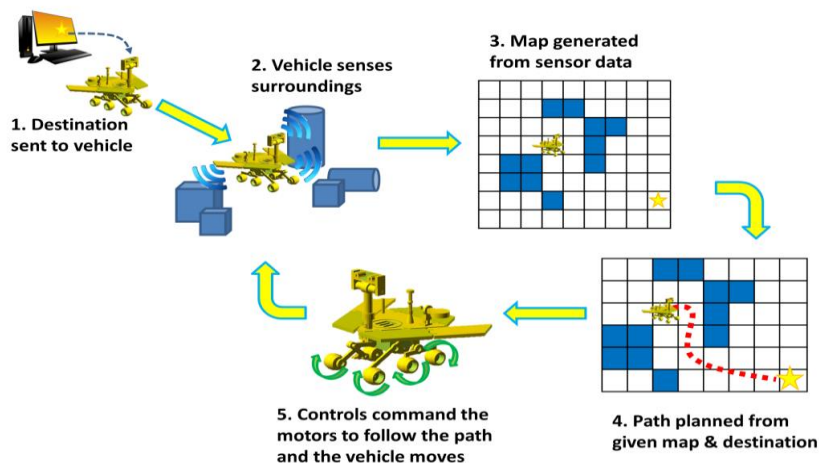
---

**Step 6:** Create a list of all of the engineering characteristics you are considering for your current concept. While the performance criteria you have come up with are more general and can be applied to any potential solution, the engineering characteristics are more specific to your current concept, often referring to both functional and structural aspects of your system design.

Continuing our example, below is a list of possible engineering characteristics for our example autonomous vehicle. Before going over that list, for those who might be unfamiliar with autonomous vehicles / robots, below is a brief description and figure on how our current concept system operation is being thought of.

1. The operation begins with the vehicle getting a command to go to a new destination (the star) and operate its payload and navigation sensors along the way.
2. To perform this task, the vehicle has a suite of on-board navigation sensors that help it to estimate the world around it...
3. ...and populate a local area estimate “map”, i.e. a grid that says how safe it is to travel in each cell.
4. This map is then used by a path planning algorithm to determine a route for the vehicle to travel from its current location to the desired destination.
5. The vehicle then runs a low level controls algorithm to create commands for the vehicle’s motors to drive the vehicle along the calculated path.

As the vehicle moves, new sensor information is brought in to update the map and the process repeats itself.



**Figure 1: Example of Autonomous Vehicle Operation**

From this description, it becomes easier to notice some potential engineering characteristics. Some of the more obvious ones listed below were hinted at earlier, like the maximum speed. Sensor accuracy and how good our system is at interpreting the sensor data into something meaningful are also characteristics of our system that also become apparent as having a meaningful impact on our potential performance – it's going to be hard to navigate if we don't have a reliable picture of our surroundings. Going further, the frequency at which the map is updated, the frequency the path planning algorithm is re-run to account for any map updates or new destination commands, and how often adjustments are made to the motor commands are all characteristics that could have significant impacts but may not have been ones we would have thought of at first. Similarly the precision and accuracy of these last three aspects will also have a significant performance impact, and all three may also vary depending upon how we design our system.

Other aspects of our system in the list include that our vehicle might have different kinds of sensor ports, and a different number of each type. To operate those sensors we might need different microcontrollers (smaller, possibly more specialized electronics boards) and a main computer onboard the vehicle. In the list below “quality” is a general term used where the higher quality an engineering characteristic is made, the more it is assumed to provide better results but may take more computational resources or require fancier components to operate.

1. Maximum Speed
2. Maximum Acceleration
3. CPU
4. # Microcontrollers
5. Number of Sensor Ports Type A
6. Number of Sensor Ports Type B
7. Number of Sensor Ports Type C
8. Navigation Sensors Quality
9. Sensor Filter Quality
10. Area Map Precision
11. Path Planning Quality
12. Path Planning Frequency
13. Controls / Path Following Accuracy
14. Controls Quality
15. Structural Material Costs
16. Structural Repair Costs
17. Electrical Parts Costs
18. Electrical Repair Costs
19. Initialization & Calibration Time
20. Assembly Time
21. Manufacture Time
22. Frame Strength
23. Max Payload Weight
24. Max Payload Size
25. Sensor Bay Size
26. Battery Capacity

There are likely to be even more engineering characteristics for a system like this one. Similarly, this QFD process can be applied to far simpler systems as well (it's been commonly used in toy, and even toothbrush design too!). However, the complexity of our example here will help in later Steps to demonstrate the QFD's ability to highlight beneficial relationships and trade-offs

between performance criteria and engineering characteristics as well as between different engineering characteristics.

**Step 6 is complete when:** you have generated your initial list of engineering characteristics for your current concept and entered them into the QFD template.

Overall, a good practice is to come up with as many as you can, not worrying about having too many, or whether they are too broad or specific. It's natural to refine them further as we go through later Steps – possibly combining a few under a new name, or splitting some into several new ones.

---

**Step 7:** Setup the QFD to discuss how adjusting each engineering characteristic will affect each performance criteria. This is a remarkably easy step and can't be done incorrectly! (well, maybe if you tried really, really hard you could).

Looking at the QFD example Step 7 tab, you'll notice that a row called "direction of change" has been added below the engineering characteristics. In this row, in the cell under each engineering characteristic is an arrow pointing up or down. If the arrow points up, this simply means that when talk about adjusting this engineering characteristic, we are asking the question what impact will *increasing* this engineering characteristic have on our performance. For example, since the arrow under "maximum speed" is pointing up, we'll be asking the question in the QFD what impact will increasing the maximum speed have on the system's performance criteria.

We'll be answering that question in the next step but for now just add a "direction of change" arrow below each of your engineering characteristics.

**Step 7 is complete when:** You have added a "direction of change" arrow below each of your engineering characteristics.

As a typical rule of thumb, the arrow is usually chosen to be point in the direction that is at least initially believed to have an overall positive effect on your system's performance. It doesn't have to be though. That's just a common practice.

---

**Step 8:** As you can see in the example QFD, you've created a matrix in the middle of the QFD with performance criteria rows and engineering characteristic columns. Now we'll be filling in those matrix cells (often referred to as the QFD's "main body" or "main floor") with an estimate on how adjusting each engineering characteristic will affect each performance criteria. This is where things start to get exciting! – Or at least more meaningful.

But first we have to go over some of the traditional ways to fill in the cells – one of which is with an integer rating from -2 to +2. Positive numbers indicate t

hat adjusting the engineering characteristic according to its direction of change arrow will have a positive effect on that performance criteria. For example the direction of change for the Maximum speed engineering characteristic was given an up arrow in Step 6. Therefore we now ask what will affect increasing the Maximum speed will have on each performance criteria. (See the example's Step 8 tab). For example, increasing the Maximum speed will have a very positive effect on the time to accomplish tasks, so that cell is given a "2".

Traditionally, instead of writing numbers in the cells the set {-2,-1,0,+1,+2} is represented symbolically often with pluses for the positive numbers, checkmarks for the negative numbers and the cell's with a zero are left blank, i.e. {xx, x, , +, ++}. It is also common to use checkmarks

instead of pluses for positive numbers. Not quite as common but still useful is to sometime use a High, Medium, Low rating which is often represented with zero still as a blank as {-H, -M, -L, , L, M, H} which is the numeric equivalent of {-3,-2,-1,0,+1,+2,+3}. Other symbolic representations are frequently used as well but the general meaning is always the same.

Going to wider scales than a -3 to +3 rating is strongly recommended *against*. Many groups might even shun a -3 to +3 rating, accepting only a +2 to -2 rating. Why? The truth is that especially when the QFD is typically first being used in a design process, the design concept is not flushed out well enough nor perhaps are the engineering characteristics influence on the performance criteria understood well enough to justify that much of a distinction. Is something that might be given a “5” really five times more influential than something that is given a “1”? That’s not something we can usually defend well at an early stage.

Instead at this point we’re trying to establish general trends. This is part of the reason why symbols are commonly used instead of numbers – we’re trying to represent with the “x” and “+” that we feel an engineering characteristic either strongly (xx or ++), somewhat (x or +), or not that significantly (zero) affect the performance criteria. Consequently, many times this part of the QFD is rather sparse with most cells being given a value of zero because although there may be some small influence, the vast majority of the time we really don’t have enough information to “split hairs” and accurately quantify that small influence, so we just leave that cell blank.

**Step 8 is complete when:** You have assigned an estimate on how adjusting each engineering characteristic will affect each performance criteria.

We’ll investigate the results of filling out these cells in later Steps, but if this is your first time using a QFD, we highly recommend using the widely used standard of {xx, x, , +, ++} or {-2,-1,0,+1,+2} or {-2,-1, ,+1,+2}. When using symbols, it is also common to include a symbolic legend which has now been added in the example to Rows 10-15 of Columns A and B.

To later make your life easier, there is also a formula to convert symbols to numbers (or the reverse) in the Calculated Imputed Importance Tab which is discussed in Step 12.

---

**Step 9:** Estimate how adjusting each engineering characteristic will affect each other engineering characteristic. This is obviously very similar to the last Step but now we’re beginning to recognizing new trade-offs within our design concept itself.

For example, increasing the Maximum speed engineering characteristic could be said to have a negative effect on the battery capacity because increasing the Maximum speed will most likely cause a larger drain on the batteries. Similarly, improving the CPU engineering characteristic would have a positive effect on many of the “quality” engineering characteristics as a better CPU will make it easier to operate these complex functions.

Either of these could be recorded in the reverse way as well, i.e increasing the maximum battery capacity could have a positive effect on the maximum speed since there would be more stored energy available. Hence, the “QFD roof” is shown as just one half of a matrix as the complete matrix would be a symmetric triangular matrix (i.e. identically mirrored across its diagonal).

In a few cases, you’ll notice that the relationship between the engineering characteristics is not symmetric. For example, increasing the Maximum speed would make developing the path planner and the controls more difficult (i.e. a negative relationship). However, improving the path planner and the controls would allow the maximum speed to be increased (i.e. a positive



relationship). In these special cases, the asymmetric relationship is written as the column engineering characteristic's effect on the row engineering characteristic, then a “/”, and then the row engineering characteristic's effect on the column engineering characteristic.

If there is an asymmetric relationship between engineering characteristics and in one of the directions there is thought to be a neutral relationship, the neutral relationship is represented as a zero or sometimes a “-“. This is shown in the example where improving the Sensor filter quality will have a positive effect on the Controls quality, but improving the Controls quality is thought to not have a significant effect on the Sensor filter quality.

**Step 9 is complete when:** You have assigned an estimate on how adjusting each engineering characteristic will affect each other engineering characteristic.

---

**Step 10:** Take a moment to review the information we've assembled so far. We're not going to make any decisions yet though. In the next few Steps we'll use this information to calculate some summarizing values which will help to offer more perspective to consider, as well as help us establish our design targets. But there still is enough information to improve our understanding of our current concept.

As a simple example of what to look for, consider the following: As hinted at in the beginning of this document, making something go faster might initially sound like a great way to improve the top performance criteria “time to accomplish tasks” and perhaps even the “coolness” performance scores. However, we also now formally recognize that achieving a faster speed will likely decrease your “safety” performance.

Then there are other performance effects that might not have been as obvious. Going faster might require the vehicle to require a more structurally sound frame in order to achieve the same performance. Alternatively, in order to go faster, this might require a lighter total body design that negatively influences your maximum payload.

Similarly, to go faster many of your other engineering characteristics may also need to be adjusted. If you're going faster, your sensors now might have to be able to collect data more quickly, which may now have to be analyzed more quickly, and perhaps the accuracy of their overall interpretation might also be affected. Going faster may also require more power, which in turn could influence your power system's engineering characteristics by requiring more or larger batteries to achieve the same runtime performance. More batteries in turn could influence the weight of your system, which in turn could influence how fast you are able to go – which was what you were hoping to increase in the first place! Whoa... These kinds of ripple effects and considerations that must be taken from making what at first appeared to be a great simple change is a very good example of the kind of understanding we're seeking to find.

**Step 10 is complete when:** At the very least you have begun to recognize at this stage how interrelated the design of your system's components really is. Similarly just going this far with the QFD should help your team recognize the importance of setting up some interface specifications – i.e. if say there's a subteam that is deciding how fast the vehicle should move, they're going to need to check with a number of other groups on how their speed decision is going to affect the other groups as well.

There are many interface tracking tools out there, but a simple one that at least demonstrates the important kinds of information to track can be found in the Interface Tracking Guide. We're still going to do some more work though with the QFD to help us set design targets.

---

**Step 11:** Take a break. It takes some real effort to do a QFD well.

If you haven't already, also take some time to discuss your findings so far with your teammates. Make sure that you're representing all of the engineering characteristics relationships properly and in a way that you can all agree on -- both with regards to the performance criteria and other engineering characteristics. Did you miss any engineering characteristics? Are there any other interesting observations/interactions your teammates noticed? Are you able to begin specifying interfaces?

The key thing to get out of this work so far is a better understanding of how adjusting various engineering characteristics will influence both the overall system performance and whether adjusting one engineering characteristic might also influence others. It is rarely a straightforward as we might hope or as it may initially seem – that's why we use the QFD process to help us organize our thoughts and tease out this information.

**Step 11 is complete when:** You've allowed some of the insight you've begun to recognize from the earlier Steps sink in. You're now ready to investigate further and eventually begin to make some design target decisions.

---

**Step 12:** Calculate a representative single number as to how much influence each engineering characteristic has, both positive and negative, on your potential solutions overall performance. This single number is called the "imputed importance" and is a pretty simple calculation.

1. To begin, simply transform your QFD main body symbols into equivalent absolute value numbers, i.e. {xx, x, , +, ++} becomes {-2, 1, 0, 1, 2} and then taking the absolute value becomes {2,1,0,1,2}. The result is sometimes called the "number of marks number", or simply the "marks number".
2. For each engineering characteristic, multiply each of its marks numbers by the performance criteria's weight of that row.
  - a. Example 1: The relationship between Maximum speed and Time to accomplish task is "++" which translates to a "2" mark number. The weight of Time to accomplish is 13.51%, so 2 times 13.51% gives a resulting value of 27.02%
  - b. Example 2: The relationship between Maximum speed and Navigation task is "-" which remembering we use the absolute value here translates to a "1" mark number. The weight of Navigation is 10.81, so 1 times 10.81% gives a resulting value of 10.81%
3. Add up all of the resulting values from the previous step and this gives you the imputed importance – TaDa!

Since the imputed importance treats both positive and negative marks the same, the imputed importance can give you an idea as to how critical an engineering characteristic is to your system overall. In Step 20 we'll use this and some other information to directly help create our design targets. – See I told you this where it starts to get exciting.

In the example, the symbols in the QFD have been converted to numbers for easier calculation and there is an additional tab to aid in this and in the calculation of the imputed importance.

**Step 12 is complete when:** You have completed the imputed importance for all of your engineering characteristics. These results are recorded in the QFD underneath the bottom of the QFD main body. This area, which we'll be adding to in the next few Steps is often referred to as the QFD "basement".

Strangely enough the imputed importance is often represented as a percentage -- but a quick math check says that since all of the performance criteria weights sum to 100% , if all of the QFD main body cells for an engineering characteristic had 2 marks, the maximum imputed importance would be 200%! Wow, that's really important! ;) Actually labeling the imputed importance as a percentage really doesn't make complete sense but that is still the common way to write these things. Oh well.

---

**Step 12a:** There is also a positive / negative imputed importance that can be calculated. This is done in the same way but this time you are calculating two numbers for each engineering characteristic: one that is each engineering characteristic's weighted sum of their *positive* marks, and the other that is each engineering characteristic's weighted sum of their *negative* marks., i.e. just use all the positive marks only, and then just count all the negative marks only.

For completeness, the steps are written below for the negative imputed importance but the same can be done for the positive marks

1. To begin simply transform your QFD main body negative symbols into equivalent absolute value numbers and ignore all of the positive symbols, i.e. {xx, x, , +, ++} becomes {-2, -1, 0, 0, 0} and then becomes {2, 1, 0, 0, 0}
2. For each engineering characteristic, multiply each of its marks numbers by the performance criteria's weight of that row.
  - a. Example 1: The relationship between Maximum speed and Navigation task is "-." which remembering we use the absolute value here translates to a "1" mark number. The weight of Navigation is 10.81, so 1 times 10.81% gives a resulting value of 10.81%
3. Add up all of the resulting values from the previous step and this gives you the negative imputed importance – TaDa!

**Step 12a is complete when:** You have computed the positive / negative imputed importance in the QFD basement. Typically this is recorded with the positive imputed importance value and then the negative imputed importance value in the same cell, separated with a slash.

You may begin to notice some interesting results, such as the imputed importance of Maximum speed which is very large, but it actually has a larger negative imputed importance than a positive imputed importance! So does this mean we should reduce our max speed as much as possible?! These imputed importance values can bring up some very interesting questions but let's not draw any conclusions until we take a few more steps.

---

**Step 13:** As a set up to recording design targets and a couple other Steps, write down the units that be used to discuss each engineering characteristic. Units are important and need to be consistent!

Some items like "mean time between failures" are unitless. In these cases, it's usually best to at least write "NA" or sometimes written "N/A" which is short for "non-applicable". This way any other readers of your QFD know that you didn't just left it blank because you forget about it.

Sometimes for unitless engineering characteristics, like those that called some kind of "quality", the name of a Table that has an index of values/options for that engineering characteristic is included instead. This table can be written below the QFD, in spreadsheet separate tabs, or in separate document. An example of a Table like this is shown in the example for "path planning quality" as Table U1 which is written below the QFD. Here, a number of possible algorithms that

are being considered are listed in order of increasing computational capability, and in this case, also likely complexity.

Algorithms and computer programs are sometimes ones that are difficult to establish agreeable units on. Alternative “units” used are algorithms’ Big “O” value, lines of code, memory usage, or the clocked runtime on a standard hardware setup, say perhaps an Intel Edison running a specified version of Linux. There are other unusual units that may arise but the important thing is that whatever units you use, in this Step you are establishing a standard for communication – which you can imagine will be very important when you’re later establishing your interface specifications as mentioned in earlier Steps,

**Step 13 is complete when:** all of your engineering characteristics have established units or have been labeled as non-applicable.

---

**Step 14:** Whenever relevant, list the engineering characteristic values for each of the competitors you identified in Step 4. For example, what is the Maximum speed of each competitor? Be sure to list your answer in the units established in Step 13.

If the competitor does not have a relevant engineering characteristic (say perhaps a competitor solution that does not move would not have a maximum speed) simply write “NA” or “N/A” for non-applicable. In some cases, you may want to offer some explanation – it may seem weird that a solution does not have maximum speed. Hence, for those situations you may also see people write a reference number to an explanation, with the reference number written in brackets (e.g. [1]) and the explanation written below the QFD, in a separate tab, or in a separate document.

To obtain competitor engineering characteristic values, it’s fairly common to buy a competitor and test it. If this is not feasible, due to costs or other reasons, an estimate should at least be used. Common sources to find estimates can be a competitor’s data sheet or sometimes third-party reviews, even on-line forums. To distinguish between confident values and estimate values, sometimes estimates are written in italics. In some cases, some people may even assign a confidence rating. This is less common but when it is done a regular practice is to first write the engineering characteristic value and then the confidence rating in italics, separating the two numbers with a comma.

**Step 14 is complete when:** you have the listed the current engineering characteristic values for all of your competitors.

---

**Step 15:** In many cases, you cannot set the design target to just any desired value because there may be additional constraints and requirements as to what values can be allowable. Perhaps there is a government regulation, or your customer or boss simply told you a minimum or maximum allowable value, or it’s in a pre-signed contract. Whatever the source, record these values into the QFD just below your competitor engineering characteristics.

To distinguish the two, minimum thresholds are sometimes underlined, and maximum thresholds are either left as plain text or are sometimes written in bold.

**Step 15 is complete when:** you have written down any constraint or requirement values you have for your engineering characteristics. If there are no constraint or requirement values for an engineering characteristic, simply write “NA” or “N/A” for non-applicable.

---

**Step 16:** Add an estimate on the technical difficulty it takes to work on an engineering characteristic. This can be assessed in many ways and sometimes several different technical difficulty values (each in their own row) are listed with various focuses like the ones below. Other times a 1-5 (or other scale) score is assigned based upon some combination of all of these, similar to the way a 1-5 score might be assigned in a performance metric. Possible focuses to consider together or separately include:

- The technical difficulty to develop a solution the first time
- The technical difficulty to change the engineering characteristic once the initial work has already been done
- The expertise level that is required
- The amount of labor time that would be needed
- The kind of equipment that will be necessary to use

Regardless, of how the technical difficulty value is established (even if it's just a very rough guess for your first attempt), technical difficulty will be used later on to help establish your design targets. For example, you may want to significantly increase the quality of the path following control system, but this might be a very technically difficult engineering characteristic. Hence when it comes time to establish your design targets, the technical difficulty measure should help you consider what you can practically accomplish.

**Step 16 is complete when:** a technical difficulty score has been assigned to each engineering characteristic.

It is common to try to assess the technical difficulty on the tasks directly associated with that engineering characteristics and not the ripple effects on other engineering characteristics. For example, for increasing the Maximum speed you may want to consider only the time & effort it takes to select new motors. The potential ripple effects can also be handled in the optional Step 23. However, you are all the more welcome to include whatever factors you think best in the assessing your technical difficulty of your engineering characteristics, so long as you are consistent in your assessment methods across all of the them.

---

**Step 17:** Repeat Step 16 but this time making an estimate of the cost for each engineering characteristic. The cost here is typically meant to be primarily financial as you have considered other needs such time and expertise in the technical difficulty estimate.

Similar to technical difficulty, cost may also have several possible focuses to consider together or separately which can include:

- Fixed costs
- Variable (i.e. non-fixed) costs
- Labor costs
- Materials Costs
- Equipment costs (such as having to pay for time on a special machine)
- Testing Costs (every time something is changed, it may have to go through a validation process again)
- The rate at which any of the above focuses change as you change the engineering characteristic

Like technical difficulty, if this is your first attempt you may want to just use a single number estimate and then formalize things later on as you get more data on the above aspects. Even a rough estimate is better than nothing when you're starting out.

---

**Step 17 is complete when:** a cost score has been assigned to each engineering characteristic

---

**Step 18:** Re-evaluate your engineering characteristics and performance criteria relationships given your technical difficulty and cost estimates for your engineering characteristics.

For example, we might want to try determining how much it would cost to increase the Structural integrity to say a slightly better or thicker material. If Structural material cost and Manufacturing time is inexpensive enough and technically easy for us do, we may want to reduce the influence Structural material cost & Manufacturing time may have had on other engineering characteristics. Continuing the example, Maximum Speed is shown in the QFD roof to have a negative relationship with Material cost and Manufacturing time. But since we see Structural material cost and Manufacturing time are “cheap & easy” may decide to reduce the negative relationship influence between Structural integrity and the Maximum speed.

As another example, seeing the strong negative effect between Maximum speed and some of the path and sensor quality engineering characteristics which have significant technical difficulty, we may want to increase the influence the Maximum speed has on Navigation or Measurement Accuracy.

The goal is to make sure you have the best representation of how each engineering characteristic influences the performance metrics to make good design decisions early on.

**Step 18 is complete when:** you have made any changes to your engineering characteristics and performance criteria relationships you think necessary. If any changes are made, re-compute the affected imputed importance scores as well.

---

**Step 19:** Take a break! Phew, you’ve done a lot. We just have one more major step to take and that’s determining our design targets but that will take some real focus to do well.

You will also notice the shape the QFD matrix is rather unique. We’ve been referring to many of the QFD parts with names like the “roof”, “back porch”, or “basement” and that’s because some believe the QFD matrix kind of looks like a house – hence the QFD is frequently referred to as the House of Quality. The house metaphor names of the QFD matrix are summarized on the Step 16 tab.

**Step 19 is complete when:** you’re feeling refreshed and clearheaded enough to make some key decisions coming up next.

---

**Step 20:** Determine your first design targets estimates! Yes, it’s now time! Even with all this great information, this will still take some serious thought.

Here’s a common tip for helping you to monitor the effectiveness at setting your design targets. As you set your first design target estimates, it is common to re-compute your overall performance score in the QFD back porch based upon how a systems operating at your design targets would perform. Some people keep their original estimates and add in a new column(s) for their new iteration(s) as they create different possible combinations of design targets and what to estimate what the overall benefit of that approach will be.

As one tried and true way begin,

1. Start with any estimates for your design targets that you think best represent your current concept that you began the QFD process with.

- a. If you don't have estimates, or simply find that you no longer like your current concept estimates, a common practice is to start with your minimum performing constraint/requirement engineering characteristic values, i.e. what must a minimum solution be able to do.
  - b. Or if there are no constraint/requirement values, you can use a realistic number inspired by your competitors' engineering characteristic values.
2. Starting with no more than your top **5** engineering characteristics with the *largest* imputed importance, adjust their target values using your positive / negative imputed importance scores to try to improve your estimated overall performance.
- a. Consider the technical difficulty and cost to figure out what it takes to adjust engineering characteristic knob and how much time and resources you might want to dedicate to this engineering characteristic.
  - b. For example, if the direction of change arrow is pointing up and the positive imputed importance is greater than the negative imputed importance, common guidelines for initial adjustments include increasing the target value:
    1. until your maximum performance score from adjusting just this one engineering characteristic is reached.
    2. to be as large as you can technically accomplish and can afford (recognizing you can't spend all of your time & budget in one place)
    3. at least as high as your top competitor
    4. the median value of your competitors
3. Once you have a rough idea on possible design target values for your top engineering characteristics, start to address your others in small batches of roughly 5 or less. You can continue to do this on imputed importance but may want to consider those of highest technical difficulty or cost as well at this point.
- a. It is also not uncommon to have to re-adjust some of your top engineering characteristics' target values.
  - b. Be sure to update your performance scores accordingly

By following the vehicle example, we can now revisit that interesting result in the positive / negative imputed importance of the speed which says that we actually want to decrease speed to improve our overall performance! Obviously, a zero max speed vehicle may not be the wisest nor practical solution. So instead, we may choose to begin with setting the design target at the lowest max speed that we're comfortable with -- perhaps the lowest of our competitors or the lowest that our constraints / requirements will allow.

This still sounds somewhat counter intuitive and will be addressed further in Step 22. Remember that the QFD body, which is where imputed importance is calculated from, provides an overview of relating engineering characteristics to performance criteria. As we investigate other engineering target values and the influence that engineering characteristics have on other engineering characteristics (the QFD roof) we may (or may not) recognize a desire to increase the max speed to make it easier to improve other engineering characteristics and thereby increase the overall performance.

**Step 20 is complete when:** you have established your first estimate of your engineering characteristic design targets and you have an updated estimate on your potential system's performance.

---

**Step 21:** Iterate on your QFD and design targets. You may have naturally done some iteration in completing Step 20. However, it's often worthwhile to consider a few more perspectives for your overall design process planning. One classic way to iterate on your QFD is to consider what your design targets might be if you're not able to achieve the level of technical ability that you currently intend.

For example, what if you are unable to achieve one of the more complex algorithm implementations? What if a part, like the sensors, doesn't operate as well as you anticipated? What if you were not able to achieve any one of your engineering characteristics design targets? Would you be able to, or need to, place more effort into one of your engineering characteristics? Would you have to change the target for those as well? What would this do to your overall performance? These are important questions to consider as they can help you recognize how critical one engineering characteristic is, even if it may not have had as large of an imputed importance.

The above argument helps us consider technical difficulty related challenges and their effects, and the same kind of questions should also be asked with regards to cost related challenges as well as any other factors that could be outside of your control. What if your budget gets cut? What if a critical part comes in late? It's far better to understand how your design might be affected before the issues arise so you can plan accordingly, rather than have the issue come up and just "wing it". Any kind of "knee jerk reaction" typically looks at the problem in too narrow of a view. – That's one of the reasons why Step 17 above recommends considering no more than 5 engineering characteristics at a time. Even those of us who are experienced in an area, as human beings, we are typically only really good at weighing about 5 options at a time.

Another common truth to consider when iterating on your design targets: Optimal solutions commonly occur when you have pushed your design to the maximum that your constraints will allow – i.e. there is typically some design factor that prevents you from achieving any greater performance. The QFD helps you to recognize how different engineering characteristics relate to one another and hence the constraints on one may help set the limit on the others. In the robotic vehicle example, we see that we may want to push another engineering characteristic(s) to its limit instead of the max speed.

Often though to find which constraints are truly the most limiting, it takes many attempts to iterate on the design. Perhaps we'll want to go with pushing the path planning or path following to be its best and then go as fast as we can adequately control the vehicle. Or we may find that that potential payload sensors, or even our available budget for motors, will require us to go slower than what our best control algorithm will allow.

It sounds like the kind of a problem that would be nice to have a computer program optimize around. However, optimization strategies are only as good as the information that we put in, and we've used a lot of estimates in our QFD. We as humans can handle estimates far better than most computer programs and even estimates help us to build intuition.

There are efforts in Systems Engineering to develop computer modeling techniques to optimize any design in this way, but as you can imagine the models are time consuming to build and again only as good as the information that we put into it. Additionally how do we model pushing the path planning to its "best" when we haven't determined what path planning strategies to consider yet? Perhaps we can model our performance as a deviation from optimal but then we'd need to know what the optimal is like -- which may also be difficult to determine. There are still a lot of challenges to be solved in System Engineering design tools like this, but for large complex systems that are very expensive, or simply operate in situations where the failure of the system



is too severe (space systems, military systems, even infrastructure and health systems) designers will take the time to create more complex system models to improve their design targets. But toy & toothbrush designers and new novel space craft & life-saving medical device designers alike, they all at least use the QFD, or QFD-like, tools to build system intuition, estimates for where to place resources, and establish design targets to help ensure they are creating one of the best performing solutions that they can.

**Step 21 is complete when:** You have developed the best design targets and estimate of performance that you currently can. It is important to revisit the QFD regularly as you continue to develop your solutions and learn more about both your solution and its challenge's needs.

---

**Step 22:** Resolve any remaining misunderstandings you had about your current concept or the challenge you had overall. We all enter any design situation with our own unique backgrounds and experiences. Some of that experience will aid us, some of it will hinder our ability to see what is actually occurring and the trade-offs and interfaces that are inherent in our current challenge. The QFD offers a formal process to help us question ourselves and allow us to re-define our understanding of a challenge. It is important to recognize these differences not only to help us move forward with our current design but to help us better approach new challenges in the future.

For example, why do we have the natural feeling though that we really want the vehicle to go really fast! Part of the reason may be that the Maximum speed has a very positive relationship to the most important (most heavily weighted) performance criteria "Time to accomplish tasks". It is common that we, as humans, naturally tend to seek to optimize around where we see the greatest gain, often considering only a few factors. In doing so may naturally ignore the importance of other factors, and in many cases result in an overall suboptimal, potentially even invalid solution, even though we "feel" that it's great and we may even feel that we can defend why – "Well of course this is a great solution! Look how good it does at our top performance criteria!"

What makes this situation even more challenging is that Maximum speed also has a strong influence on the "Coolness" performance criteria. We as humans are naturally more susceptible and give more weight to "coolness" – otherwise marketing wouldn't work. Hence we may be fighting our own intuition as we not only want it to be able to do everything that it's supposed to but to be really cool in the way it does it – then we can show it off to our friends and they'll realize that our work is great even if they know nothing about our challenge's needs, right? Overall we tend to over allocate resources to those aspects of the project that address our personal needs & desires. However, we are rarely designing something to meet our needs -- even if we're an example of the target audience, overall the design team is likely to be only a very small sampling of that audience. Instead we have to constantly remind ourselves of the needs of our challenge and how the performance of any solution will be evaluated as a whole. When you make this realization you can also begin to look at every design as its own kind of game on how to best position your effort to gain the highest score. Just make sure you're scoring your design by that challenge's performance criteria and not just your own ;)

You want to know what the best a part of this robotic vehicle example is? This example is not just contrived to help make an important point. This example was taken from part of a larger QFD that was done for a real world project – and yes, aiming for the system to run just above the minimum required speed proved to be a very wise decision. Developing many of the systems for the real robotic vehicle, like the autonomous navigation particularly the path following control system, turned out to be very complicated. Had a faster system been designed, it would have made these subsystems even harder to develop and it is unlikely that a working

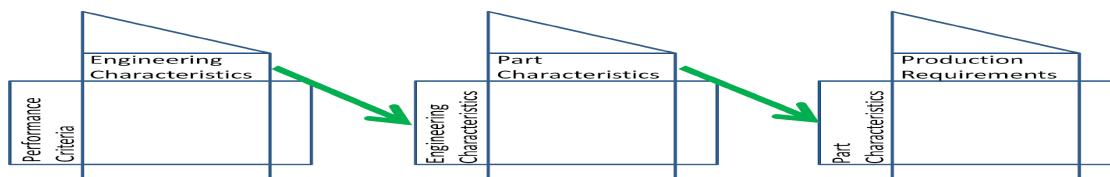
solution would have been achieved in the time allowed. Instead, the team was able to develop a system that was easier to test and learn from and met all of the challenge's needs well – enough so that they earned a rather substantial contract to continue their work further. The team was then able re-iterate on the design later to make it faster but was able to do so with a better understanding of the supporting subsystems and challenge's that would arise from going faster.

Coincidentally, the performance listed for “Competitor C” in the example QFD is the actual performance achieved by the

**Step 22 is complete when:** you have recognized why various initial ideas you had may or may not be as beneficial to your system as when you first began the QFD process.

---

**Step 23:** (optional) The QFD process is one that is very well recognized and well used, and hence there have been numerous variations that have been developed over the years. One of the most common variations is referred to as a “Linked House of Quality”. As depicted below in Figure 2, a linked house of quality is actually a series of QFD matrices where the second floor of the earlier QFD, becomes the front porch of the next QFD. So in the Figure 2 example, in the second linked QFD we are now relating how the system parts we are considering affect the engineering characteristics. Using our vehicle example, a possible part might be the motors we're considering, which obviously have an influence on the Maximum Speed and Maximum acceleration engineering characteristics but the motors most likely influence a number of other parts such as the power and controls circuitry and the motor mounts and vehicle chassis.



**Figure 2: Example of the general format for a series of linked QFDs**

The direction of change arrow in later linked QFDs can sometime be a little tricky to interpret and you may need to include a note to indicate what it means. A more intuitive example might be, if the direction of change for the motors points up, in making the above comparisons you would be considering the effects of changing to larger more powerful motors. A more challenging part might be a sensor. But in this sensor case you might make a list of possible sensors and the direction of change arrow refers to moving up or down that list. In some cases, a direction of change arrow can be harder to specify for a single part but maybe it's possible for a part of part or a collection of parts (i.e. specific set of components on the power board).

In Figure 2, the third QFD matrix then compares the selected parts to the production requirements, for example, how might various tolerances, or buy vs. build decisions influence the parts.

This is just one example of how to link together QFD matrices, but the overall idea is that by linking the QFD matrices you can now see how changing the tolerance of a single part could trace back to making an impact on the performance criteria. Regardless of whether you use QFD matrices to help make connections like this or not, it's always important to recognize all of the influences that various design decisions could make on system at all levels, especially on the overall performance.

**Step 23 is complete when:** You have considered other ways of using the QFD process to help you address other aspects of your overall design process.

---

**Appendix A: Example Engineering Characteristics Explanations:** If you're unfamiliar with robotics or uncertain about the meaning of any of the engineering characteristics, below is some basic background on the engineering characteristics to help you better understand the example.

- **Maximum Speed** is how fast the robot can travel. **Maximum Acceleration** is how quickly the robot can achieve that speed from a stopped position. **Maximum Acceleration** also relates to how quickly the robot may be able to change direction.
- The **CPU** is the main processing unit which handles many of the more complex computations such as interpreting the sensor data and running the more complex algorithms like the path planning and possibly some of the higher level controls. **Microcontrollers** handle simpler computational tasks and the direct input and output commands to other components, such as reading in the sensor data (example of an input), or aiding in sending of the proper voltage to motors so they will actually move (example of an output). Some **microcontrollers** may be able to do some initial filtering of sensor readings to improve the value of the sensor data. **Microcontrollers** may also run some simpler lower level controls programs such as helping to make sure the motor wheels are spinning at the desired speed. Determining that desired speed or what to do with those improved sensor readings is handled more often by a **CPU**.
- **Sensor Port Type** refers to the idea that different sensors have different connection requirements. Some sensors may require more input/output pins on a microcontroller, some may require specific communication protocols, etc. For this example, the QFD is considering 3 different kinds of sets of sensors requirements. The details for each set is not important for this example, but it is worthwhile to recognize that each sensor type may affect the performance criteria differently.
- **Navigation Sensor Quality** relates to how accurately the robot can detect its surroundings. As the quality of these sensors increase, it typically means that the sensors are larger and/or more numerous and also require more computational support to operate and interpret.
- **Sensor Filter Quality** is a measure for how much processing the raw sensor data undergoes. Typically, the more filtering the better the quality (i.e. the more useful & reliable) the resulting processed data is. There is often a diminishing return for the amount of effort put into the sensor filtering and the quality of the resulting processed data, i.e. it's often easy to filter away some of the larger noise but to get better accuracy it takes significant effort to tune the filters and/or develop algorithms to refine sets of data.
- **Path Planning Quality** is how good of a path does one create, where goodness could be defined by criteria such as minimizing the path length's distance or maintaining a safe distance from certain obstacles. **Control / Path Following Accuracy** is how well do you follow the path you have created, independent of the speed you are travelling along the path. **Controls Quality** is related to **Control / Path Following Accuracy** but **Controls Quality** refers to how well your system may be able to react to quick changes in

commands, or maintain a desired output disturbances (e.g. keep going straight when going over a bump)

- **Material & Parts Costs** are the cost of the components to build the robot in the first place. **Repair Costs** are the costs associated with fixing the robot and may also include an estimate of the time / labor required to perform the repair along with the direct costs of the components.
- **Initialization & Calibration Time** refers to the time it takes to prepare the robot and all of its sensors, algorithms, etc. for use when brought into new operating conditions or otherwise has been repaired, significantly modified, or reset.
- **Assembly Time** is the amount of time it takes to put the robot together for the first time. **Manufacturing Time** is the amount of time it takes to machine, construct, and otherwise ready all of the components for the robot to be put together.
- **Frame Strength** refers to the amount of force the robot can withstand before enough damage has been done to affect the robot's performance.
- **Max Payload Weight** refers to the maximum amount of weight in addition to that of the robot that can be carried without altering the robot's performance to be outside of its requirements. **Max Payload Size** refers to the space available to securely carry a payload. **Sensor Bay Size** refers to the space, separate from the payload space, that is dedicated to accommodating additional potential on-board sensors.
- **Battery Capacity** refers to the size and energy storage (which are proportionally related) of the on-board batteries.