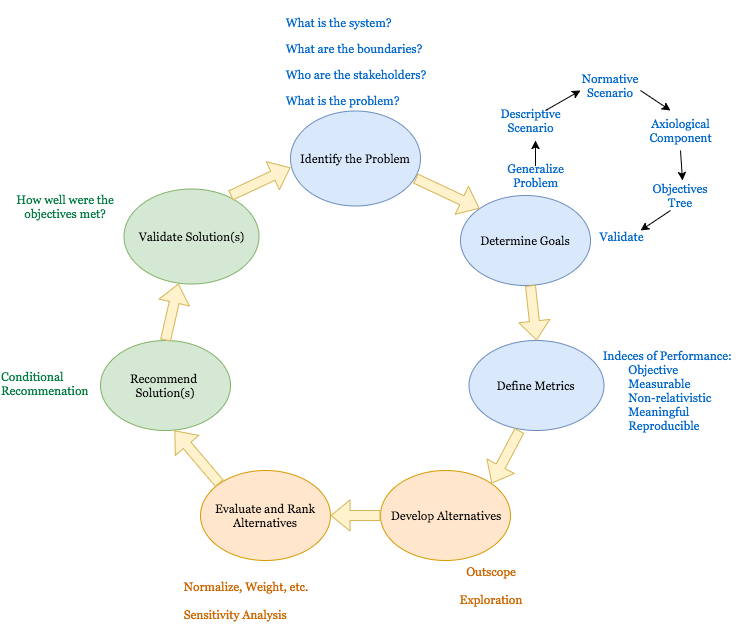
**The Systems Engineering Approach**

Jessica Krynitsky



Systems Engineering methodology is a nonlinear, robust approach to solving large scale, complex problems. I believe this methodology can be broken down into three general stages, as illustrated by the figure on page one: problem exploration (blue), solution development (orange), and validation (green). I have separated more discrete steps within these stages, which are very similar to Gibson and Scherer’s Six Steps studied this semester. In the following text, I walk through these stages, using the experiences I have had this semester as illustrations of the information I provided. Some sections are more detailed than others due to time constraints.

**Identify the Problem**

It is easy to jump straight to solutions as soon as a problem is presented; this is something the human brain does automatically. The task of the systems engineer is to pause and before even suggesting a type of solution, think about what the problem is in context. In the Amtrak case, we were inexperienced system engineers and I remember the first thing my group talked about were ideas of ways to improve the system, but we artificially limited ourselves by not broadening our scope of thought beforehand. Some important questions that must be answered: What is the system? (Is it the train and the track, the Amtrak network, the country’s transportation system?) What are the system’s boundaries? (Is ATC considered? Does the system include the physical geography or the political context?) Who are the stakeholders? (What stake do passengers, conductors, Amtrak owners, the government, etc. hold in this problem?) All of these questions and more must be answered in order to determine the problem and the context in which it exists.

**Determine Goals**

After the problem and system are defined, they must be investigated more closely in order to determine the goals of the system. The steps to this process are illustrated as another iterative sub-process. Generally, a systems engineer must look at what the situation looks like now (descriptive scenario), how it should look (normative scenario), and then quantify that change into a set of objectives. It has been interesting for me to discover how true and necessary this process is to solving any problem, and I have seen myself using it loosely in my Computer Science class as well as personal decisions. Additionally, I noticed that during pre-departure meetings for the Argentina study abroad program, we collectively have practiced techniques synonymous to those we learned in this class.

**Define Metrics**

Defining metrics has been one of the most

**Develop Alternatives**

It is important that the previous steps do not limit the options for alternatives that can be considered. An interesting example of this that I found is that NASA found that earthworms can survive in Martian soil.1 It is easy to see that this discovery must have come about from a broad set of objectives. It is also indicative of the importance of diversity of thought in engineering. If NASA scientists were only Aerospace engineers, as some uninformed people seem to believe, it is unlikely that any of them would have had the backgrounds in Chemistry and Biology needed to make such a discovery. This gets back to the idea that systems engineering projects require diversity of thought to be most successful. I have noticed that the best group dynamics on cases this year occurred not when each of us thought in the same way, but when we all filled different roles and challenged each other. I noticed this most strikingly in the final bridge case.

**Evaluate and Rank Alternatives**

Once alternatives are developed, you use the Indices of Performance that you developed to come up with a ranking for the alternatives. This seems practical, but it is amazing how many decisions are made without any concrete value.

**Recommend Solution(s)**

The systems engineer is not the decision maker. Our task is to develop the best alternatives that meet the objectives we have defined, and then present those to the client or person making the decision. An important step here is identifying trade-offs that inevitably exist, and covering these trade-offs in a way that is thorough enough that all the decision-maker has to do is decide based on his values. It is important that these alternative recommendations account for variables, uncertainty, and ways the solution could change over time. The George’s Tshirts case offered a terrific learning experience for this.

**Validate Solution(s)**

This is the stage in which you determine how well your objectives were met by the solutions you found, and then iterate.

**Iterate**

Unlike Gibson, I did not include “iterate” as a formal step in my process. This is because I have found that iterating is necessary throughout every aspect of the project. I depicted this by making my diagram a cycle, which never reaches a natural end. In reality, the process would look more like a web of arrows pointing to every step, which follows the general order I have given. This was clearest to me in the final case of recommending bridges to VDOT. My team found ourselves reevaluating what the problem was, what was included in the system, and what the system objectives were several times before we even came up with a ranked list of bridges. This happened again when we thought about how to perform sensitivity analysis. We asked ourselves, “We could figure out a way to do the same sensitivity analysis we did for the transportation case (case 4), but is the problem similar enough? Would a completely different type of analysis be more appropriate for our problem? Are our objectives accurately defining what the system needs to accomplish?” We spent almost as much time asking these questions as we did programming and crunching numbers to come up with a solution. This is integral to systems engineering.