Some problems are so unique that they require the engineer to invent completely original solutions. Most of the time, however, the engineer can solve even complex problems by building the system from components that already exist. Creativity will still be required in selecting the proper components, making small changes in their behavior (tweaking), arranging them in an effective and efficient manner, and then verifying the system satisfies both the requirements and constraints. When young engineers begin their first job, they are sometimes surprised to see that education does not stop with college graduation, but rather is a life-long activity. In fact, it is the educational goal of all engineers to continue to learn both processes (rules about how to solve problems) and products (hardware and software components). As the engineer becomes more experienced, he or she has a larger toolbox from which processes and components can be selected.

The hardest step for most new engineers is the first one: **where to begin**? We begin by analyzing the problem to create a set of specifications and constraints in the form of a requirements document. Next, we look for components, in the form of previously debugged solutions, which are similar to our needs. Often during the design process, additional questions or concerns arise. We at that point consult with our clients to clarify the problem. Next we rewrite the requirements document and get it reapproved by the clients.

It is often difficult to distinguish whether a parameter is a specification or a constraint. In actuality, when designing a system it often doesn't matter into which category a parameter falls, because the system must satisfy all specifications and constraints. Nevertheless, when documenting the device it is better to categorize parameters properly. **Specifications** generally define in a quantitative manner the overall system objectives as given to us by our customers.

Constraints, on the other hand, generally define the boundary space within which we must search for a solution to the problem. If we must use a particular component, it is often considered a constraint. In this class, we constrain most designs to include a Tiva LaunchPad. Constraints also are often defined as an inequality, such as the cost must be less than $50, or the battery must last for at least one week. Specifications on the other hand are often defined as a quantitative number, and the system satisfies the requirement if the system operates within a specified tolerance of that parameter. **Tolerance** can be defined as a percentage error or as a range with minimum and maximum values.

In engineering everything is either a system or an interface between systems. For example a switch can be considered a system. When we interface it to the LaunchPad the switch-LaunchPad combination is a new system. Therefore, we begin by collecting the components required to build the system. We then combine the components and debug the system. As the components are combined we create new more powerful components. When writing software, we can use flowcharts to develop new algorithms. The more we can simulate the system, the more design possibilities we can evaluate, and the quicker we can make changes. Debugging involves both making sure it works, together with satisfying all requirements and constraints.

Specification: Design two systems that include a switch and an LED. The analog system includes just the switch and LED, and the digital system adds a microcontroller. The switch controls whether an LED is turned on/off using negative logic. In particular, if the switch is pressed the LED is off, and if the switch is not pressed the LED is on.