The Essence of Engineering

UFI -- www.ufiservingscience.com
10-2016

This is a Blog post, and we have engineering here. And there seems to be only a vague understanding about what engineering really is. And not everybody has or can have the 'tool-set' to be an effective Engineer.

1. An initial comparison: Scientists vs. Engineers

a. Scientists discover then carefully prove or disprove things. Scientists deal with facts, experimentally finding out facts, what is true, and disproving what is false. Their focus is justifiably narrow, and often focused on one area of knowledge. Scientists deal with knowledge, prove or disprove knowledge, typically one fact at a time.

b. Engineers apply knowledge to solve real-world problems. An Engineer's knowledge must be much broader than the narrow focus of a Scientist, though perhaps not as deep. And many problems that Engineers solve require the pulling together of many facets of knowledge from a usually broad array of disciplines.

2. The Essence of Engineering

The Engineering process, what an Engineer does, is like a pyramid, whose vast size supports the actual engineering solution resting calmly at it's peak. This pyramid has the following layers.

a. Knowledge and Understanding

Knowledge forms the huge base of this pyramid. Engineers have to know a lot. Engineers have to remember a lot. Engineers must have immediate access to facts from science, spread often across a painfully broad array of topics. This memory pool includes both short term and long term memory. For Engineering, both will need to be huge, and quickly accessible.

But it is not enough to simply 'know' a lot. Memorizing the dictionary will not make a good Engineer. Every fact in our data base has a unique plug on it. Understanding refers to how we correctly use each fact. And Engineers need to understand, to grasp that plug, how that fact gets used, very quickly!

b. Analysis

The not quite as huge middle section of that pyramid is analysis. An Engineer absolutely must be able to quickly grasp and understand 'interactions' between the facts involved. A particular situation / requirement / mechanism / etc. must quickly fan out into a series of interactions between facts, laws, behaviors, etc, *all in the Engineer's mind*. Maybe that is why good Engineers love to take stuff apart! It could be a co-worker's subroutine, or someone elses' schematic, or the exploded view of that rolling press, or the worn out coffee machine in the back. Taking it apart, we are actually analyzing, identifying all the interactions and basic science involved. Engineers smile when they do that! The analysis capability shows up more brightly in another major area -- trouble-shooting. Fixing broken things kicks the analysis function into high gear. You often need to carefully understand everything about the broken thing, and mentally bring together even more interactions in order to find out why it has stopped working! Hopefully you can fix it, or at least you understand why it can't be fixed.

c. Synthesis

This step is the heart and soul of Engineering, a smaller slice near the top of the pyramid. When you make a bead necklace, you grab some thread, then pull beads from a bunch of different little boxes, and put them on the string, one at a time. When you are done, the bead necklace is both pretty, it fits, and maybe says something intelligable. That is a poor picture, but Engineering works like that. Engineering looks at the requirement, the real world problem, then pulls together facts, mechanisms, code, circuitry, ETC, out of sometimes hundreds, maybe thousands of little boxes, keeping in mind any manner of interactions (temperature, time, motion, etc.) all from a potentially wide array of disciplines, and then carefully weaves everything together into a solution to the problem. As the Engineer works on the solution, EVERYTHING needs to stay in his head, in it's proper relationship and orientation between all the rest of the pieces, facts, mechanisms, code segments, ETC. And, if some new arrangement needs to be made, the Engineer figures out how to make that work, again based on the facts, mechanisms, code segments, interactions, ETC, between well, everything else in the solution. Larger systems may require breaking the problem down into sub-problems, each handled by an individual Engineer. This doesn't change much, except that the interfaces between the sub-problems need to be clearly defined, and the Engineer still needs to know how his sub-solution fits in with the rest.

Whether you work with code, hydraulics, circuits, mechanism, bridges, or anything else that requires a solution to a real-world problem, this is Engineering. There will probably be a lot of discussion about how much you can teach, how much of this you can actually learn. But, I know this much. Not everyone is 'wired' to take an old toaster apart, or enjoy walking through a junkyard, or working through TRS80 code, or climbing down inside an old ship, or (add your own here)... Some of this is just who we are.