The Devil is in the Details:
Why Engineering is an Inexact Science

Day W. Radebaugh
Assistant Professor of Philosophy
Department of Philosophy
Wichita State University
Wichita, Kansas 67260

Abstract

One of the main reasons cited for the choice of an engineering career is the desire to employ an exact science. Most students of engineering distinguish themselves from the practitioners of "soft" sciences, such as psychology, marketing, sociology or philosophy. After all, the so-called soft sciences have no findings which are certain, but rather competing sets of assertions that arise from one or another set of values.

This paper argues that engineering in general is not entitled to the degree of certainty typically associated with it. This point is illustrated by analyzing eight topical areas of engineering design, showing the limits of certainty in each case. Further, a case is made that the recognition of this inherent uncertainty should be acknowledged and taught in engineering ethics classes. In the first instance, understanding the limits of knowledge in engineering is a useful antidote to a riskier approach which might tend to ignore unknowns. Secondly, it would seem that an objective assessment of what an engineer is entitled to claim to be certain, and what is regarded as uncertain should precede any attempt to evaluate the ethical implications of the technology under consideration.

I. Introduction

There are eight categories of decisions that must be made during the course of an engineering project. The accuracy of these eight categories of decision depend in each case on the degree of certainty of the information used to make these respective judgments.

I.A Project Issues

Every undertaking, from the small to the large, the simple to the complex, is subject to the dynamics of the project. We can associate 5 management dimensions with each project: scope (roughly, the requirements or objectives), deadline, budget, quality of the work performed, and, particularly in the case of engineering efforts, the safety of the system produced. These 5 dimensions of management form the context within which all subsequent engineering decisions must be made, in particular the remaining decision making categories discussed below. The essential problem for the engineer is that it is, in practical terms, impossible to satisfy all of these 5 constraints at once. As everyone knows, there will not be enough budget to achieve all stated objectives, or there will not be enough time to complete the project as originally described. The inevitable consequence is that the engineer must sacrifice one of the five constraints to satisfy another; for example, the quality or safety of the system is neglected to meet the deadline, or to remain within budget. Trade-offs among these objectives are made, and, to make matters worse, these trade-offs are made without full information about their ultimate consequences. For instance, when sacrificing quality to meet price objectives or market deadline, the engineer frequently does not know what the final consequences of that decision will be. In the case of the Ford Pinto gas tanks, there was perhaps relative certainty among project engineers that the vehicle would be unsafe, but there was very much less certainty associated with the decision to adopt the compromised O-ring design that led to the disaster of the Challenger shuttle.

It is important to emphasize to engineering students that the trade-offs that must be made among these considerations, and the degree of uncertainty attached to each is very typical of ordinary value-based decision we make in our non-professional life, such as compromises among lifestyles, careers, investments, and the like. The condition of uncertainty affects each of the categories discussed below.

I.B Uncertainty of Data

One would imagine that the performance of an experiment or test, and the observation and collection of data would be the most solidly and unimpeachably "technical" of the decision-making exercises an engineer must perform. However, this process is not nearly as objective as one might suppose. Any observation one might make is made in a context which includes, certainly, the external constraints discussed in connection with project constraints, but also more technical compromises and assumptions. In other words, no observation stands alone, but depends on measurement methodologies and assumptions which are often not stated. For example, we measure a temperature, and would like to regard that datum as accurate; however, its accuracy is only as good as the instrument we use, or more generally the measurement methodology employed. In most cases, one can ignore these dependencies; however, these assumptions must be considered in situations where the ethical consequences could be most significant. For example, imagine an engineer designing a new type of car structure, with the objective of achieving greater safety while saving weight and cost. Typically the engineer would design such a structure on a design workstation, and might indeed perform some testing using software models. However, the design will ultimately be built and tested at full scale. To validate this design, the engineer would instrument this structure with gauges indicating the stress on the parts during a crash. Yet the selection of points to measure, and the development of a model is precisely the type of decision which is subject various types of uncertainties. For example, it might not be altogether clear where to measure these forces, particularly in the event that the structure's design might be radically different for preceding efforts. In the final analysis, these decisions will be subject to the project considerations discussed above; in other words, trade-offs and compromises will be made in the testing and modeling because of budget, deadline, and the like.

I.C "Laws"

Once again, the engineer would like to believe that he uses strictly accurate laws in the design process. Such laws have indeed been developed over the decades and are widely accepted. There are, however, two additional rules or expectations which, although they do not have the epistemological authority of accepted empirical laws of engineering, must nevertheless be considered. Hence the use of the term "laws" in referring to these considerations.

The first of these quasi-laws is the law of unintended consequences, which, simply stated, means that any project may well produce results that could not reasonably be foreseen. For instance, what will be the effect of introducing a new biological agent into the ecosystem? We cannot in general predict these consequences, but must be prepared for their occurrence. Of course, the law of unintended consequences is a commonplace observations. Yet there is an interesting reason why this regularity will hold true, and be particularly significant, in engineering projects. Unintended consequences are often the results of scale differences in science; that is, they arise from the difficulty of applying the results of experiments on one scale to another scale. For example, engineering at the nano level (i.e., 10^{-9} meters) is often regarded as providing breakthrough solutions to some engineering challenges. We find this expectation in genetic engineering as well as non-biological applications. However, who can predict the consequences of nano-level changes when introduced into the larger ecosystem or biological chain? Such a level of uncertainty will be particularly crucial as these methodologies are more widely adopted.

The second "law" is quite familiar to all engineers, and is called Murphy's Law: whatever can go wrong, will go wrong. Although we all acknowledge this law, why is it true, and with a vengeance, in engineering projects. If we look past the commonplace, we find an interesting reason in the complexity of engineering projects. Consider an engineering system most broadly as a series of boxes or sub-systems connected by various communication systems, whether electrical, hydraulic, pneumatic, or a host of other methodologies. Each component of the whole has a range of behavior it will exhibit; that is, even in the digital domain its behavior will deviate from one instance to another. If we further regard these components as embedded in a complicated matrix of interconnections, we realize that it may be very difficult to predict how the system will respond from one occasion to the next. Under a long set of trials, we are likely to see most possible perturbations or deviations of these components, with sometimes unpredictable results, which is exactly the force of Murphy's Law. Murphy's Law, therefore, is of necessity going to hold, particularly so as engineering systems become more and more complex.

I.D Material Selection and Use

In general, the selection and use of materials is subject to the same type of uncertainties discussed above. Briefly, consider the more objective evaluation of the behavior of a new material under consideration; can the engineer be quite certain about the behavior of this material under new conditions? Further, the choice to use a material or technique is subject to the project limitations discussed above, particularly those of scope, deadline and budget. To make matters worse, the cases that will be most difficult to decide will be exactly those applications on the leading edge of engineering application. Material science is changing very rapidly, and will present many areas of uncertainty as new techniques are adopted.

II.E Software

The use of software in design and testing has constituted a revolution in engineering, enabling entirely new types of structure and systems to be built. However, increasing reliance on software leads to a new category of uncertainty. In the first place, software will only be as good as the data or input it requires; we have discussed the uncertainty of data in II.B above. However, the software itself may contribute another degree of uncertainty to the process. Can we rely on its calculations? It would be unusual indeed if there were not at least one mistake in a model's calculations. Is the model or process implemented in the algorithm regarded as accurate in the first place? Finally, there may be some processes which are so complicated in their own right that the software designer is unable to fully test the algorithm. The Star Wars missile defense system, while never fully implemented, was thought to be such a system.

II.F Testing and Reliability

Testing and reliability present some unusual uncertainties in the product life cycle. In the first case, one must decide how to test a design: in software, at small scale, or full-scale live testing. The problems of software modeling and differences of scale have been discussed above. If we test in the lab, we must realize that laboratory testing is an ideal condition. In testing for failure, the engineer is entitled to regard the failure of a component as, in some sense, a success; at least he has determined the failure point of the component. The more interesting question is when to stop testing if failure does not occur. Can we accurately estimate the duty cycle of a product? Quite often the decisions about how much to test, and when to stop testing, are subject to the usual project constraints of budget and deadline.

II.G Human Factors

Without doubt humans represent a major element of uncertainty in a product life cycle. Human users are quite unpredictable, and their interaction with a system, particularly during problem analysis and resolution, is often difficult to foresee. Moreover, engineers as a group tend to overestimate the technical literacy of the user community, and thus do not anticipate problems when a product is introduced because the design seems to be so well thought out.

II.H Complexity

Engineering implementations during the past several decades at least have been characterized by increasing complexity. This complexity, at first glance, can be described in two dimensions. The first dimension could be categorized as complexity of inherent design, and describes the underlying processes responsible for the system's behavior. The second dimension is perhaps more generally familiar, and refers to the complexity of usage or perceived complexity: the nature of the interaction of the system with its environment, either other systems or the human user itself. The reader can quite likely supply examples of interactions with everyday systems that illustrate this occasionally bewildering complexity. Part of the reason for Apple's remarkable success in introducing the iPod and iPhone can be attributed to its unique simplicity of usage.

Both types of complexity certainly increase our understanding of modern technological systems, and our subsequent ability to control their behavior. Such complexity is at least partly responsible for a number of recent engineering disasters, such as the massive outages of the electrical power grid in 1965 and later in 2003, the nuclear accident at Three Mile Island in 1979, and the various disruptions of the World Wide Web.

Biographical Information

DAY RADEBAUGH is a Visiting Assistant Professor in the Department of Philosophy at Wichita State. He has been at Wichita State since 2003, and has taught courses in Computer Ethics, Engineering Ethics, Introductory Philosophy, Assembly Language Programming for the PC (Computer Science Dept), and Military History (History Dept).