

Defining “Engineer” — How to Do It, and Why It Matters

by Dr. Michael Davis

“Today, the field [of ‘software engineering’] has emerged as a true engineering discipline.” — John J. Marciniak
“Preface,” *Encyclopedia of Software Engineering*, 1994

“If you are a ‘software engineer,’ you could be breaking the law. It is illegal in 45 states to use that title, warns *Computerworld* newspaper. People who aren’t educated and licensed in 36 recognized engineering disciplines can’t call themselves ‘engineers,’ and computer professionals often don’t qualify.” — *Wall Street Journal*, June 7, 1994, p. 1

THIS ARTICLE BEGINS with recent events in what may or may not be the history of engineering, the emergence of *software engineering* as a distinct discipline. The term *software engineering* seems to have come into currency after a NATO conference on the subject in 1967.¹ Today, many thousands of people are called *software engineers*, do something called *software engineering*, and have sophisticated employers willing to pay them to do it.*

Yet, *software engineering* is no ordinary engineering discipline. Many *software engineers* are graduates of a program in computer science, having a single course in *software engineering*. Typically, that course is taught by someone with a degree in computer science rather than engineering. While there seem to be no hard numbers, most *software engineers* are probably programmers with no formal training in engineering. Are *software engineers* nonetheless engineers?

The Standard Definition of “Engineer”

The standard definition of *engineer* looks something like this: An engineer is a person having at least one of the following qualifications: a) a college or university B.S. from an accredited engineering program or an advanced degree from such a program; b) membership in a recognized engineering society at a professional level; c) registration or licensure as an engineer by a governmental agency; or d) current or recent employment in a job classification requiring engineering work at a professional level.² The striking feature of this definition is that it *presupposes* an understanding of the term *engineering*. Three of the four alternatives actually use the term *engineering* to define *engineer*, and the other, alternative c), avoids

* “In the 1991 Computer Society [of the IEEE] membership survey, 54% of the current full members polled indicated that they consider themselves *software engineers*, as did 40% of the affiliate members.”

doing the same only by using *as an engineer* instead of *to practice engineering*.*

This definition, and those like it, are important. They determine who is eligible for admission to engineering’s professional societies, who may be licensed to practice engineering, and who may hold certain jobs. Such definitions are also eminently practical. For example, they do in fact help the Census Bureau exclude from the category of engineer the drivers of railway engines, janitors who tend boilers in apartment buildings, and soldiers wielding shovels in the Army’s Corps of Engineers. These, though still called *engineer*, clearly are not engineers in the relevant sense.

The standard definitions do not, however, suit our purpose. They will not tell us whether a *software engineer* is an engineer or even how to go about finding out. A *software engineer* may, for example, work at a job classified as requiring *software engineering* [at a professional level]. That will not settle whether she is an engineer. What an employer classifies as *engineering* (for lack of a better word) may or may not be engineering (in the relevant sense).**

What will settle the question? In practice, the decision of engineers will. One organization of engineers accredits baccalaureate and advanced programs in *engineering*. Other organizations of engineers determine which societies with *engineer* in the title are *engineering societies* and which — like the Brotherhood of Railway Engineers — are not. Engineers also determine which members of their societies practice engineering (*at a professional level*) and which do not. Governmental agencies overseeing registration or licensure of engineers, though technically arms of the state rather than of engineering, generally consist entirely of engineers. And, even when they do not, they generally apply standards (education, experience, proficiency, and so on) that engineers have developed. Engineers even determine which job classifications require engineering work (*at a professional level*) and which do not.

*Compare the more elegant Canadian definition³: “The ‘practice of professional engineering’ means any act of planning, designing, composing, evaluating, advising, reporting, directing or supervising, or managing any of the foregoing that requires the application of *engineering* principles, and that concerns the safeguarding of life, health, property, economic interests, the public welfare or the environment.” [Italics are author’s.] The report contains no definition of “engineering principles.”

**Similar problems arise for “genetic engineer” and might arise for other “engineers,” for example, “social engineers.” (This problem of definition is, of course, not limited to engineers: lawyers are no more successful defining “the practice of law,” or doctors “the practice of medicine.”)

The decision of engineers has settled many practical questions, but not ours. Engineers divide concerning whether *software engineering* is engineering (in the appropriate sense).⁴ So, to say that *software engineers* are engineers, if in the opinion of engineers they engage in engineering (at the professional level), is, for engineers and those who rely on their judgment in such matters, merely to restate the question.*

That is a practical objection. There is a related theoretical objection. A definition of *engineer* that amounts to *an engineer is anyone who does what engineers count as engineering* violates the first rule of definition: “Never use in a definition the term being defined.” That rule rests on an important insight. Although a circular definition can be useful for some purposes, it generally carries much less information than a non-circular definition; it generally conceals foundational questions rather than helping to answer them.

How might we avoid the standard definition’s circularity? The obvious way is to define *engineering* without reference to *engineer* and then define *engineer* in terms of *engineering*. The NRC in fact tried that approach, coupling its definition of *engineer* with this definition of *engineering*: “Business, government, academic, or individual efforts in which knowledge of mathematics and/or natural science is employed in research, development, design, manufacturing, systems engineering, or technical operations with the objective of creating and/or delivering systems, products, processes, and/or services of a technical nature and content intended for use.”

This definition is informative insofar as it suggests the wide range of activities which today constitute engineering. It is, nonetheless, a dangerous jumble. Like the standard definition of *engineer*, it is circular: *Systems engineering* should not appear in a definition of *engineering*. The same is true of *technical*, if used as a synonym for *engineering*. (If not a synonym, *technical* is even more in need of definition than *engineering* is and should be avoided for that reason.) The NRC’s definition also substitutes uncertain lists — note the “and/or” — where there should be analysis. Worst of all, the definition is fatally overly inclusive. While *software engineers* are engineers according to the definition, so are many whom no one supposes to be engineers. Not only do applied chemists, architects, and patent attorneys clearly satisfy the definition, but thanks to the “and/or” between *mathematics* and *natural science*, arguably, so do actuaries, accountants, and others who use mathematics to create financial instruments, tracking systems, and other technical objects for use.

Though much too inclusive, this definition of *engineering* shares with most others three characteristic elements: First, it makes mathematics and natural science central to what engineers do.** Second, it emphasizes physical objects or physical systems. Whatever engineering is,

* The IEEE has defined “software engineering” as “application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software: that is, the application of engineering to software.”⁵ This definition (or, rather, that “that is”) begs the question whether the systematic, disciplined, and quantifiable approach in question is an application of engineering to software or the application of a different discipline. Not all systematic, disciplined, and quantifiable approaches to development, operation, and maintenance are necessarily engineering. Indeed, the fact that software is primarily not a physical but a mathematical (or linguistic) system certainly suggests that engineering principles have, at best, only limited application.

its principal concern is the physical world rather than rules (as in law), money (as in accounting), or even people (as in management). Third, the definition makes it clear that, unlike science, engineering does not seek to understand the world but to remake it.*** Engineers do produce knowledge (for example, tables of tolerances or equations describing complex physical processes), but such knowledge is merely a means to making something useful. Engineers also produce beautiful objects (for example, the Brooklyn Bridge or the typical computer’s circuit board). They are nonetheless not artists (in the way architects are). For engineering, beauty is not a major factor in evaluating work; utility is.

Those three elements, though characteristic of engineering, do not define it. If they did, deciding whether *software engineers* are engineers would be far easier than it has proved to be; we could show that *software engineers* are not engineers simply by showing that they generally do not use the *natural sciences* in their work. That many people, including some engineers, believe *software engineers* to be engineers is comprehensible only on the assumption that these three characteristics do not define engineering (except in some rough way). But if they do not define engineering, what does?

Before answering that question, I shall describe three common mistakes about engineering to be avoided in any answer.

Three Mistakes about Engineering

The NRC’s definition of *engineering* uses *technical* twice, once as a catch-all (*or technical operations*) and once to limit the domain of engineering (*of a technical nature and content*). It is the second use of *technical* that concerns us now. It seems to be an instance of a common mistake about engineering, one even engineers make. We might summarize it this way: engineering equals technology. Anyone who makes this mistake will find it obvious that engineers, whatever they may have been called, have been around since the first technology (whether that was the Neanderthal’s club, a small irrigation system, or a large public building). Engineering will seem the second oldest profession.

There are at least three objections to this way of understanding engineering. *First*, engineering can equal technology only if we so dilute what we mean by *engineering* that any tinkerer would be an engineer (or, at least, be someone engaged in engineering). Once we have so diluted engineering, we are left to wonder why anyone would want a *software engineer* rather than a *programmer*, *software designer*, or the like to do software design or development? What was the point of inventing the term *software engineering*? (Note, for example, reference 6: “[The] term *software engineering* expresses the contin-

** I have charitably ignored the “or” in “mathematics and/or the natural sciences.” There has never been a time when the training of engineers has not included a good deal of both mathematics and the physical sciences (at least chemistry and physics). If *software engineers* do not generally have similar training in the physical sciences, no amount of training in mathematics will fill the gap between them and the great body of engineers strictly so called.

*** Compare the Smetonian Society’s now classic definition of “civil engineering”: “. . . the art of directing the great sources of power in Nature for the use and convenience of man.”

ued effort to put programming into the ranks of other engineering disciplines.”)

The second objection to *engineering equals technology* is that the proposition makes writing a history of engineering (as distinct from a history of technology) impossible. The history of engineering just is, according to this proposition, the history of technology. Every (successful) inventor is an engineer; every (successful) manager of industry is an engineer; and so on. We are left to wonder why our term for engineer — unlike our term for architect, mathematician, or artisan — is so recent. Why does *engineering* have a history distinct from *technology* when engineering just is technology?

The third objection to *engineering equals technology* is that it transforms talk of engineering ethics into talk of the ethics of technology. Whatever engineering ethics is, it is, in part at least, the ethics of a profession — not merely standards governing the development, use, and disposal of technology but standards governing a certain group of technologists.

That reference to *profession* suggests a second mistake commonly made about engineering, one we might summarize in this way: *engineering is, by nature, a profession*. What makes this mistake attractive is the idea that a professional just is a *knowledge-worker*, that special knowledge defines each profession (as well as the underlying occupation). Connecting profession with knowledge helps to exclude from the profession of engineering those who, though they may function as engineers (or, rather, as *mere technicians*), lack the requisite knowledge to be engineers strictly so called (“engineers at the professional level”). Claiming that engineering is, by nature, a profession provides an antidote to the first mistake, but only by making another.

What is this *second* mistake? Thinking of engineering as, by nature, a profession suggests that organization has nothing special to do with profession. As soon as you have enough knowledge, you have a profession. There could be a profession of one.

Thinking this way makes much of the history of engineering mysterious. Why, for example, did engineers devote so much time to setting minimal standards of competence for anyone to claim to be an *engineer*? Like other professions, engineering has a corporate history that such non-professions as shoe repair, inventing, and politics lack. Any definition of engineering must leave room for that history. What is striking about the history of engineering — indeed, of all professions — is the close connection between organization, special standards, and claims of profession.

A *third* mistake may help to explain the appeal of the second. We might summarize it this way: the engineering profession has always recognized the same high standards. There are at least two ways this mistake has been defended. One appeals to the *nature* (or *essence*) of engineering. Any occupational group that did not recognize certain standards would not be engineers or, at least, not be engaged in engineering. Engineers (it is said) have organized to write standards to avoid being confused with those who were not *really* engineers. The standards simply record what every good engineer knows.

The other argument for this mistake appeals to the moral nature of the engineer. In every time, it is said,

engineers have generally been conscientious. To be conscientious is to be careful, to pay attention to detail, to seek to do the best one can. To do this is to be ethical. Professional ethics is just being conscientious in one’s work. To be a conscientious engineer is, then, to be (by nature) an ethical engineer.* Engineering societies adopt standards to help society know what to expect of engineers, not to tell a conscientious and technically adept engineer what to do. Informing society is, according to this view, enough to explain the effort engineers put into codes of ethics.

What is wrong with the proposition that engineering is, by nature, ethical? Like the other two mistakes, this third makes understanding the history of engineering harder. Why have engineers changed the text of their codes of ethics so often? Why do experienced engineers sometimes disagree about what should be in the code of ethics (as well as about what should be in their technical standards)? Why do these disagreements seem to be about how engineers should act, not about what to tell society?

If we examine a typical code of engineering ethics, we find many provisions that demand more than mere conscientiousness, for example, avoiding conflict of interest or helping engineers in one’s employ continue their education. Such codes are less than 150 years old. Before they were adopted, an engineer had only to be morally upright and technically proficient to do all that could reasonably be expected. In those days, engineers had no responsibilities beyond what law, market, and morality demanded (and so, no need to inform society what to expect). The claim that engineering has always accepted the same high standards — that, for example, failing to inform a client of a conflict of interest has always been unprofessional — is contrary to what we know of engineering’s history.

Defining Engineering as an Occupation

To avoid these three mistakes, we must briefly re-examine the history of engineering. (For more detail, see reference 8.) There have been things called *engines* for a long time, but until a few centuries ago *engine* simply meant a complex device for some useful purpose. The first people to be called *engineers* were soldiers associated with engines of war (catapults, siege towers, and the like). They were not yet engineers in the sense that concerns us. They were engineers only in the sense that they operated (or otherwise worked with) engines.

As late as the seventeenth century, no European power had a permanent military of consequence. War was still largely the art of nobles who learned war from their fathers or on the battlefield. Armies were raised for a campaign and disbanded when it was over. In such armies, an *engineer* was usually a carpenter, stone mason, or other artisan bringing civilian skills to war.

When Louis XIV ended the regency in 1661, France still made war in that way. Over the next two decades, however, France created a standing army of 300,000, the largest, best trained, and best equipped European fight-

*Note that Mr. Florman⁷, generally so astute about engineering, endorses this equation of conscientiousness with ethicalness.

ing force since the Roman legions. This achievement was widely copied. To this day, most of our military terms — everything from *army* itself to *veille*, from *bayonet* to *maneuver* — are French. *Engineer* is just one of these military terms.

Until 1676, French *engineers* were part of the infantry. But, in that year, the *engineers* were organized into special units, the *corps du génie*. This reorganization had important consequences. A permanent corps can keep much better records than isolated individuals, can accumulate knowledge, skills, and routines more efficiently, and can pass these on. A corps can become a distinct institution with its own officers, style, and reputation. More than a group of proto-engineers, the *corps du génie* was, potentially, both a center of research in engineering and a training ground for engineers (in something like our sense), *officiers du génie*.

The *corps du génie* soon began realizing this potential. Within two decades, it was known all over Europe for unusual achievements in military construction. When another country borrowed the French word *engineering* for use in its own army, it was for the sort of activity the *corps du génie* engaged in.^{9,10} That was something for which other European languages lacked a word.

But, as of 1700, the *corps du génie* was not a school of engineering in our sense; it was more like an organization of masters and apprentices (“proto-engineers” of various ranks). I speak here not of the craftsmen in the corps but of the officers. Though engineers sometimes describe themselves as deriving from craftsmen of one kind or another, the way I am telling their story, they derive from military officers of a certain kind (those who commanded craftsmen). Engineering was, in other words, born an occupation of gentlemen (in modern terms, an occupation requiring at least a bachelor’s degree).

Only during the 1700s did the French slowly come to understand what they wanted in an *officier du génie* and how to get it by formal education. By the end of the 1700s, they had a curriculum from which today’s engineering curriculum differs only in detail; they had also invented engineering. Civilian engineering was just a branch on this tree.⁹

What distinguished engineers, military and civilian, from other *technologists*? Consider their nearest competitors, the architects. While engineers resembled architects in being able to make drawings for construction projects, develop detailed instructions from those drawings, and oversee the execution of those instructions, they differed from architects in at least three ways:

First, engineers were much better trained in (what was then) the new mathematics and physics than the architects. They had the ability to consider systematically questions most architects could only deal with intuitively or ignore.

Second, because the strategies of engineering had their roots in the necessities of war, engineers paid more attention to reliability, speed, and other practical considerations. So, for example, the systematic testing of materials and procedures in advance of construction was early recognized as a characteristic of engineers.⁹ At least in comparison, the architect seemed an artist, someone for whom beauty claimed much of the attention that an engineer would devote to making things work.

Third, to be an engineer was to have been trained as an army officer, to have been disciplined to bear significant responsibility within one of world’s largest organizations. Engineers were therefore likely to be better at directing large civilian projects than architects, most of whom would have had experience only in much smaller undertakings.

These three advantages in science, utility, and management tend to reinforce each other. For example, large projects not only require more planning in advance and more discipline in execution, they are also more likely to require better mathematical analysis and to justify extensive testing of materials and procedures. For this, and perhaps other reasons, engineers slowly took over much of the work that once would have been the domain of architects or various crafts.*

Early experiments in engineering education culminated in the *École Polytechnique*, founded in 1794. The *École Polytechnique*’s curriculum had a common core of three years. The first year’s courses were geometry, trigonometry, physics, and the fundamentals of chemistry, with practical applications in structural and mechanical engineering. There was a good deal of drawing, some laboratory and workshop, and recitations after each lecture. The second and third year continued the same subjects, with increasingly more application to the building of roads, canals, and fortifications and the making of munitions. For their last year, students were sent to one of the special schools — artillery, military engineering, mines, bridges and roads, and so on.¹²

Engineers will immediately recognize this curriculum, especially the four years, the progression from theory (or analysis) to application (or design), and the heavy emphasis on mathematics, physics, and chemistry. The *École Polytechnique* was the model for engineering education for much of the nineteenth century.¹³ It is startling how quickly the model was taken up. For example, just eight years after the founding of *École Polytechnique*, the military academy at West Point, our first engineering school, was already trying to become “the American *École Polytechnique*.”¹⁴ (I add that West Point largely failed at this during its first two decades. Reference 14 is good on the practical difficulties threatening the very life of the academy during its early years.)

The history of engineering education in the United States from then on has two strands: one is a series of unsuccessful experiments with various alternatives to the West Point curriculum; the other, the evolution of the West Point curriculum into the standard for engineering education in the U.S. The details of this story do not matter here. (For more, see reference 15.) What does matter is that the education of engineers came to seem more and more the province of engineering schools, and these in turn came to be more and more alike. For engineers, an engineer came to be someone with the appropriate degree from an engineering school or, absent that, with training or experience that was more or less equivalent.

* Compare reference 11: “At some point, the science becomes sufficiently mature to be a significant contributor to the commercial practice. This marks the emergence of engineering practice in the sense we know it today” As I tell the story, engineering and commerce have no necessary connection. Indeed, it is the connection with military utility that is crucial.

The point of this story is not that engineering will always have the same curriculum it does today; the curriculum has changed much since 1794 and, no doubt, will continue to change. The point, rather, is that just as today's curriculum grew out of yesterday's, so tomorrow's will grow out of today's. Any new field of engineering will have to find a place in that curriculum. Finding a place may mean changing the curriculum; what it cannot mean is starting fresh. Finding a place in a curriculum is a complex negotiation of social arrangements. It is like joining a family. You can change your name to *Davis* if you like, make yourself look like a member of my family (perhaps even genetically), and declare yourself a member of my family, but that won't make you one. To be a member, you must enter by birth, marriage, or adoption.

Some fields of engineering (for example, nuclear) seem to have been born engineering, but others (mining, for example) seem to have joined by (the occupational equivalent of) marriage or adoption. For any field not born engineering, the only way to become a field of engineering is by *marriage* (or *adoption*). Failing that, it cannot be a field of engineering. It can only don quotes or invite confusion.

Membership in the Profession of Engineering

What I have sketched so far is a history of engineering as a distinct occupation, an alternative to the mistake of equating engineering with technology. I have, however, not sketched a history of engineering as a profession. The history of a profession tells how a certain occupation organized itself to hold its members to standards beyond what law, market, and morality would otherwise demand. The history of a profession is the history of organizations, standards of competence, and standards of conduct. For engineering in the U.S., that history hardly reaches back to the Civil War.* It is a confused history because the profession was taking shape along with the occupation. Many early members of its professional societies would not qualify for membership today.

Nonetheless, we can see that as engineers became clearer about what engineers were (or, at least, should be), they tended to shift from granting membership in their associations ("at a professional level") based on connection with technical projects, practical invention, or other technical achievements to granting membership based on two more demanding requirements. One — specific knowledge (whatever its connection with what engineers actually do) — is occupational. This requirement is now typically identified with a degree in engineering. The other requirement — commitment to use that knowledge in certain ways (that is, according to engineering's code of ethics) — is professional. While many professions (law and medicine, especially) make a commitment to the profession's code of ethics a formal

* While an organization called the "American Society of Civil Engineers [and Architects]" was actually founded in 1852, its membership was almost entirely in New York City. And, like other attempts at organizing engineers before the Civil War, this one seems to have died out within a few years. The connection with the ASCE of 1867, the real beginning of engineering organizations in the U.S., is tenuous, little more than an overlap in membership and purpose. For a bit more on this, see reference 16.

requirement for admission, engineering has not (except for licensed P.E.s). Instead, the expectation of commitment reveals itself when an engineer is found to have violated the code of ethics. The defense, "I'm an engineer, but I didn't promise to follow the code and therefore did nothing wrong," is never accepted. The profession answers, "You committed yourself to the code when you claimed to be an engineer."¹⁷

Attempts to understand *software engineering* as engineering have, I think, generally missed this complexity in the concept of the profession of engineering. Consider, for example, these observations of Mary Shaw (a professor of computer science, a *software engineer*, and the daughter of an engineer):

*"Where, then, does current software practice lie on the path to engineering? It is still in some cases craft and in some cases commercial practice. A science is beginning to contribute results, and, for isolated examples, you can argue that professional engineering is taking place."*¹⁸

Substitute "applied science" for "engineering" in this passage and there is little to argue with. But, as it stands, the final sentence of the passage is simply false. There is nothing in what Shaw describes to suggest that "professional engineering is taking place." There is nothing either about an historical connection with engineering (whether birth, marriage, or adoption) or about a commitment to engineering's code of ethics.

Conclusion

The question to be asked, then, is not whether *software engineers* are engineers. Clearly, while a few are, most are not. The question is, rather, whether (or when) they *should* be. There is no fact of the matter when identifying engineering disciplines, only a complex of social decisions in need of attention — especially, decisions about how to train *software engineers*. ¶

Michael Davis is a senior research associate in the Center for Study of Ethics in the Professions and an adjunct associate professor of philosophy at Illinois Institute of Technology, which he joined in 1986. He earned three degrees in philosophy — his B.A. in 1965 at Western Reserve University and his M.A. and Ph.D. in 1966 and 1972 at the University of Michigan.

Author or co-author of five books and numerous articles and papers, he taught courses in philosophy and ethics during 1972-86 at Case Western Reserve University, Illinois State University, and the University of Illinois at Chicago. A consultant and winner of several fellowships, since 1987 he has served as editor of *Perspectives on the Professions*.

References

1. Gary A. Ford and James E. Tomayko, "Education and Curricula in Software Engineering," *Encyclopedia of Software Engineering*, v. I, John Wiley & Sons: New York, 1994, p. 439.
2. National Research Council, Committee on the Education and Utilization of the Engineer, *Engineering Education and Practice in the United States*, National Academy Press: Washington, DC, 1985, p. 36.
3. Canadian Engineering Qualifications Board, *1993 Annual Report*, Canadian Council of Professional Engineers: Ottawa, 1993, p. 17.
4. "A Debate on Teaching Computing Science," *Communications of the ACM*, vol. 32, no. 12, 1989, pp. 1397-1414, especially those contributors who (like David Parnas) are (by training) engineers.
5. Fletcher J. Buckley, "Defining software engineering," *Computer*, vol. 26, no. Aug., 1993, p. 77.
6. L. A. Belady, "Forward," *Encyclopedia of Software Engineering*, vol. I, John Wiley & Sons: New York, 1994, p. xi.

7. Samuel C. Florman, *The Civilized Engineer*, St. Martin Press: New York, 1987, p. 105.
 8. Michael Davis, "An Historical Preface to Engineering Ethics," *Science and Engineering Ethics*, vol. 1, no. 1, 1995, pp. 33-48.
 9. Frederick B. Artz, *The Development of Technical Education in France, 1500-1850*, MIT Press: Cambridge, MA, 1966, pp. 47-48.
 10. W. H. G. Armytage, *A Social History of Engineering*, Faber and Faber: London, 1961, pp. 96 and 99.
 11. Mary Shaw, "Prospects for an Engineering Discipline of Software," *IEEE Software*, vol. 7, no. 6, 1990, p. 17.
 12. Reference 9, pp. 154-155.
 13. Reference 9, pp. 160-161.
 14. Peter Michael Molloy, *Technical Education and the Young Republic: West Point as America's École Polytechnique*, Brown University, Dissertation, 1975.
 15. Lawrence P. Grayson, "A Brief History of Engineering Education in the United States," *Engineering Education*, vol. 68, 1977, pp. 246-264.
 16. Edwin T. Layton, Jr., *The Revolt of the Engineers*, Press of Case Western Reserve University: Cleveland, 1971, pp. 28-29.
 17. Michael Davis, "Thinking like an Engineer: The Place of a Code of Ethics in the Practice of a Profession," *Philosophy and Public Affairs*, vol. 20, no. 2, 1991, pp. 150-167.
 18. Reference 11, p. 22.
-

Engineering Futures? — Plan It!

As you chapter officers begin planning your Tau Beta Pi year, remember to include time for an Engineering Futures session. Engineering Futures is a unique program offered by Tau Beta Pi which provides our members with the opportunity to sharpen their interpersonal skills and, consequently, improve their effectiveness in working with people. The sessions focus on aspects of working in teams, applying creative problem solving to a wide variety of situations, running meetings efficiently, and relating to individuals in a positive, non-punishing manner.

Each of the sessions is interactive — which means it is FUN! The sessions are facilitated by volunteer Tau Beta Pi members who are now in the workplace and realize the value of these skills! The material has been tailored especially for Tau Beta Pi from a program offered to many of the Fortune 500 companies.

Session materials and facilitator expenses are paid for by the Association, so there is no cost to the chapter, unless you want pizza! Sessions can be offered for your whole chapter, for initiates, or as a service project for the engineering school under the sponsorship of your chapter. If you are interested in more details, call 423/546-4578 for "host chapter" information.