The Engineer of the Future
University of Illinois, College of Engineering
5 Sept 07

Outline for remarks
William A. Wulf

The topic of this workshop is "The Engineer of the Future."

I'm going to break my remarks into two very different topics:

The first is related to a pair of studies done by the National Academy of Engineering (NAE) with the common title "The Engineer of 2020."

The second is some very personal speculation that goes out beyond 2020 and may be totally bogus.

<u>Context #1</u>

I went to the University of Virginia (the second time) in 1988, after (with Anita) founding and running a small software company. I was also on the board of a mid-sized civil engineering company at the time.

On returning to academia, I was struck by two things:

1. The curriculum was almost identical to the one I had experienced 30 years before, and
2. that curriculum bore almost no relation to what I had experienced in my own company, or at the civil engineering company.

I devoted the next few years to revising the computer science curriculum at Virginia, but did not do anything about the larger engineering curriculum.

When I went to the NAE, I decided that I had a bigger bully pulpit from which to tackle the larger issue.

I'll tell you a bit about what we've done at the NAE later.

<u>Context #2</u>

"They say" that where you stand depends on where you sit, so I need to say a bit about where I have been sitting the last 11 years—at the National Academies.

It's a private, non-profit corporation, BUT it also has an 1863 Congressional Charter that in some ways makes it seem schizophrenic. It has two distinct personalities:

1. Honorific
2. Advisors to the nation—telling truth to power

NAS -> {NAS, NAE, IOM, NRC}

The NAE is known for its independent, authoritative advice.

The process typically is—

Vet the question

Form a committee—

the best minds, working pro bono for 18 months on average, to produce a balanced report.

Reports are fact-based—no opinions.

Reports are peer reviewed.

All reports are available on the web. (See—NAP.EDU.)

A report of about 200-pages is produced every working day.

OK, so where I've been sitting is a policy shop—we advised other people to do things; we didn't do anything ourselves.

The NAE isn't an operationally oriented organization. We are set up to advise other people on what to do; not to do anything ourselves.

But we have a lot of prestige.

So, we are trying, with a number of specific activities, to change attitudes among your faculty colleagues to make them more receptive—and to make them accept that change is, indeed, needed.

Changing attitudes is not easy, and it will not happen overnight, but it's a role that, perhaps, only the NAE can play.

In particular, I believe the clear and consistent messages from the NAE are that:

1. The NAE believes change is <u>necessary</u>, and
2. the NAE <u>values</u> contributions to that change

and will, over time, change faculty attitudes.

To that end, we have taken four concrete actions:

1. We have created the Committee on Engineering Education (CEE).
2. We have made contributions to engineering education a valid criteria for election to the Academy.
3. We have established the $500,000 "Gordon Prize" for innovations in engineering and technology education.

4. We have established a Center for the Advancement of Scholarship of Engineering Education (CASEE) at the NAE.

The CEE conducts studies and holds workshops on topics in engineering education; some of you have seen the pair of reports called "The Engineer of 2020"—these were done under the oversight of CEE.

It would be inconsistent to say that the NAE values contributions to engineering education—but that's not enough to be elected as a member—so we changed that! People now <u>can</u>, and <u>have</u> been elected because of their contributions to engineering education.

The Gordon Prize now recognizes contributions to engineering education on a par with engineering innovation as recognized by the Draper and Russ Prizes—all are $500,000 prizes awarded by the NAE!

CASEE has a number of programs (totally about $4m this year) to both do scholarship on engineering education and to promote high quality scholarship by others.

I have to say that work on engineering education has not always had as high a standard of scholarship as engineering research—and CASEE aims to fix that.

We also are trying to bring to that scholarship the results of, for example, the cognitive sciences, as they apply to pedagogy.

It turns out that there is a great deal that has been learned in the last ten or so years about both the psychology and physiology of learning that has a direct impact on pedagogy.

We can teach more, faster, and with better learning outcomes.

Let's go back to the 2020 reports,

Over the years I've been in a number of quite unproductive discussions of the engineering curriculum.

Everybody agrees that we should teach only the fundamentals … however, there is not universal agreement on what the fundamentals <u>are</u>!

So, we decided on a two-step process:

Step 1: Use scenario-based planning to get an idea of what engineers will be doing in 2020.

Step 2: Since an engineer in school today will be in mid-career in 2020, use the result of step 1 to decide what we ought to be teaching today.

The committee came up with a number of recommendations—

1. Whatever else, start teaching the essence of engineering, (ie; iterative design), from day 1.
2. Embrace scholarship on engineering education.
3. Exploit the flexibility of EC2000.
4. Stop treating the BS as the first "professional degree" ("pre-engineering").
5. Encourage more domestic students to go on for MS & Phd degrees.
6. Promote Public Understanding of Engineering (PUE).
7. Help improve K-12 STEM education (take responsibility).

The committee didn't come up with specific curricula suggestions, but I think the pair of reports is very useful and should stimulate some creative experiments.

<u>OK, now I'd like to fly off on my own flight of fancy</u>—which as I said may be totally bogus.

I'm a computer scientist by training.

I am vaguely sheepish about the fact that no non-trivial program has ever been written that was bug-free.

From time to time I've had other engineers look down their noses at me. If we were REALLY good engineers, this wouldn't happen.

Well, I can tell you it's not for lack of trying.

1. The '68 NATO (Garmisch) Conference introduced "Software Engineering"—

   Get rigorous; use the "engineering method"!

2. We don't talk a lot about the "engineering method", but we all somewhat understand that it is a relatively linear process:

   a. Requirements analysis
   b. Specification
   c. Prototyping
   d. Testing the prototype (and cycling if needed)
   e. Production implementation
   f. Testing (and cycling if needed)
   g. Deliver

   h. Maintenance

3. In the heady days of the 70s some SE's thought we could do <u>better</u> than traditional engineers [explain formal specs & proofs]

4. It hasn't worked out that way for a number of reasons:

   a. Proofs are too big/hard—no computationally feasible way to automate them (we tried!)

   b. The specs are sometimes (often?) wrong
      i. NRL security analysis 22/50
      ii. Cryptographic protocols—there's no way to specify that "nothing *bad* will happen"

   c. Traditional "testing" doesn't work given the complexities—
      i. discrete vs. continuous systems
      ii. magnitude of the state space

   Despite CMM (rigorous certification of the process), we <u>still</u> have bugs!

5. OK, what's the underlying reason that it hasn't worked out? Here's where the speculation & possible "bogusity" starts:

   <u>Conjecture #1</u>: The real problem is that the traditional "engineering method" doesn't work for really complex systems!

   It just so happens that it's easier to build really complex systems in SW than in HW, so we're seeing the effect there first.

6. Or, is it? Consider:
   a. damming the Mississippi => building in unsafe places (like New Orleans)
   b. building "city" interstates => sprawl & decline of mass transportation
   c. deliberate introduction of "invasive" species, and
   d. network research => "globalization"

7. In each case, with 20-20 hindsight, it's easy to see a causal relation—but people didn't!

   Would they if they had thought about it harder?

   Maybe, but there are so many possible consequences that it's not clear anyone could have identified the particular consequence in advance.

Go back to the cryptographic protocols example. These are cases where the code is small, and the consequences of failure are high—so people have thought about them really hard. They still have unexpected holes!

8. So why did I bring this up? It's because I think all engineered systems in the future will exceed the threshold of complexity where we are unable to anticipate all their behaviors <u>because:</u>

    a. almost everything will have embedded SW, or
    b. will interact with the natural environment—an example being the Everglades
    c. will be based on "systems biology".
    ….

9. So, the "Engineer of the Future," I think, is going to need a different "engineering method."

    I have no clue what that method is, but…

    I brought this up to (hopefully) get other people thinking about it.