Workshop on Reshaping Doctoral Education
CRA Conference at Snowbird 2000
10 July 2000

Remarks by Clinton W. Kelly, III Ph. D
Senior Vice President
Science Applications International Corporation

- My remarks are based on an insular view reflecting trends and perceived needs in our company.

- Since we are not a household name let me begin by briefly describing the company:

  - Founded in 1969
  - Employee owned, not publicly traded
  - $6B in revenue
  - 313th on the Fortune 500 List
  - We are a scientific and technical services company
  - We essentially don't have products

- We have about 41,000 employees, about 2,200 Ph.Ds and one hundred of our Ph.Ds are CS Ph.Ds. We have about 1,000 whose highest degree is the MSCS and about 1,800 with the BSCS only.

- Our Ph.Ds are highly diverse, reflecting the diverse character of the company: Anthropologists, Botanists, Zoologists, lots of Physicists of all kinds, Geneticists, Veterinary Pathologists, Mathematicians, Seismologists,

even an Archaeologist or two, and various kinds of Engineers (EE, ME, CE, Aero, Chemical).

- The interesting number is how many CS Ph.Ds we have hired over the past three years. The number is five. The reason is that our business is changing and the skills of the CS Ph.Ds have not changed with us.

    - What's the nature of the change? Go back to 1988 when I joined the company. Almost all of our work was research. Today it is systems integration. We still do research but it has dropped significantly as a percentage and in absolute value.

    - Remember, we don't do products so we do very little activity based research or opportunistic research that you might find in a product company.

    - We focus almost exclusively on solving customer problems, i.e. doing what they will pay us to do. Some, fewer than in past years, will pay us to do research. But the majority of the problems that they care most about today are solved through systems integration; and, if you want a one-line description of what SAIC does today, that's it.

- Some examples:
  - Fingerprint identification system for the FBI.

- Health care information system for all DOD hospitals worldwide.
- The OSS for Level III, a new packet-switched CLEC.
- B-to-B extranet with info assurance and guaranteed QOS.
- Re-insurance trading system.
- Several voice over IP systems.
- An autonomous land vehicle.

♦ So, what do we need to win projects like these?
- We need system and software architects who know technology and who are both creative and critical thinkers.
- People who can creatively combine COTS products; hardware and software in ways that differentiate us on performance and cost from our competitors.
- Five, six, seven years ago we would write maybe a million or so lines of code for an integration project, relatively little COTS in the deliverable.
- Today, we may write only 100,000 lines of code which integrates 15 to 20 COTS products. Customers mandate this approach to lower their risk. Very challenging tradeoffs to select the products and design the architecture: the OS (could be NT, Solaris, Linux), most systems contain a DBMS (Oracle, Cybase, Active), you have GUI products – so 5-10 products already and you haven't gotten to the applications.
- Then for big, distributed systems (and more and more are) where should the processors be? Is the database distributed? How do you maintain it and update it?

And how about the COMS networking? And then you still have to select the hardware.

- We need people who can take a systems perspective, do the performance and cost trades and use simulations to convince themselves and the customer that we have the best solution; even use simulations to validate the design. This will typically require doing some benchmark experiments on candidate products – measuring timing and other parameters.
- But again, we need people who are creative and who can think critically – all our competitors have access to the same COTS products as we do – its how good you are in selecting the 'right' products, how clever you are combining them into an architecture, and how efficient you are in integrating them that makes a winner.
- To really do this well requires not just strict technical skills but people who can rapidly learn enough about the customer's business to properly interpret the stated system requirements. So we need people who are both motivated to self-educate about a substantive domain outside their field, and people who have the interpersonal skills to have conversations with the customer; to find out what's really on their mind. We need people who want to know, and do know, more than one thing.
- Some of the technical challenges (and thesis and research opportunities):
  - Designing with stable interfaces which allow you to bring in improved products so the system isn't obsolescent in 8-10 months.

- Designing in information assurance, availability, reliability, maintainability, life cycle cost, all very important to our customers!
- Must do this in design, you can't retrofit the 'illities'.
- Developing cost and schedule estimates where the majority of the software is COTS.
- How do we re-use parts of an architecture, or re-use specs?
- And usability. Serious concerns from our customers that massive capital investments in IT are not providing commensurate gains in productivity, the issue may be ease of use; customer paying for features they find hard to use. This is more than traditional 'knobs and dials' human factors.
- IV&V: increasingly difficult as we add intelligence to our systems.

♦ So what could you do for us? Create a Ph.D program for system and software architects. Create a curriculum that looks outward not inward. We don't care what goes on in the box or in the chip. Such a curriculum could have four emphases:

1. Continue to develop the analytic, and research skills and, in particular, the critical thinking expected of a Ph.D. We need those skills.
2. Use system-level case studies drawn from various system applications (lot of opportunities within your universities) and built around simulations to teach people how to work and think at this level. Bring in some COTS products and teach them what to look for

in benchmarking and how to do it. Emphasize system security, privacy and integrity – trustworthy systems, emphasize system availability, reliability, usability.

3. Provide them with clinical experience.
4. Require them to read and discuss broadly outside their field and encourage them to continue to read as part of their continuing education. Two reasons: first, they must understand the external forces that drive their customers. Second, I think, but can't prove, that it enhances creativity. Not so much "out of the box" thinking as "thinking in a bigger box". Look at the bookshelves of people you think of as creative, I bet you'll find they have a more interesting collection. It certainly seems to be strongly correlated around our place.

- There is probably no single answer to developing the kinds of people we need: other options might involve retraining some of our existing Ph.Ds; perhaps using a mixture of webcasting and on-line materials. Or perhaps we could work with you to set up a program where BS graduates go into industry for two years, an extended internship, then return for the Ph.D.

- I'm not suggesting that everyone be a system or software architect but from my perspective there are more opportunities now for people with these skills than those of the more traditional CS Ph.D. But note – If you want to adopt this as a major thrust you may also have to reshape the faculty – an issue is having faculty who

convey attitudes consistent with or supportive of a Ph.D. career in industry.