PREPARING STUDENTS FOR PROFESSIONAL PRACTICE

Thomas B. Hilburn, PhD, IEEE-CS CSDP
Professor Emeritus, Software Engineering
Embry-Riddle Aeronautical University (ERAU)
email: hilburn@erau.edu
web: http://faculty.erau.edu/hilburn

EMBRY-RIDDLE
Aeronautical University

An Educational Challenge (for SwE)

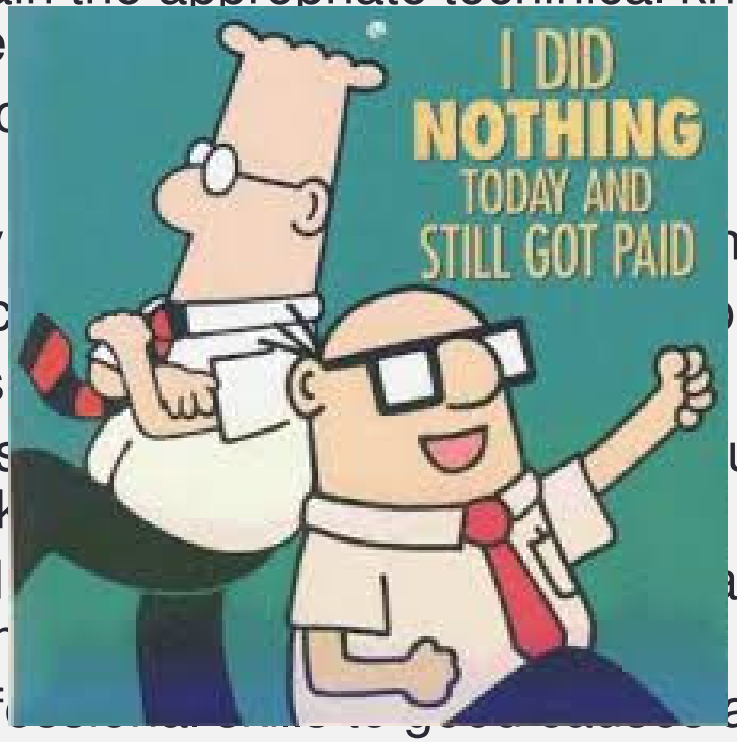
- In 1992, Peter Denning [Denning 1992] wrote

"Employers and business executives complain that graduates lack practical competence. Graduates, they say, cannot build useful systems, formulate or defend a proposal, write memos, draft a simple project budget, prepare an agenda for a meeting, work on teams, or bounce back from adversity; graduates lack a passion for learning. They say the current concepts-oriented curriculum is well suited for preparing research engineers, but not the practice-oriented engineer on which their competitive advantage increasingly depends."

What is Professional Practice?

Professionals

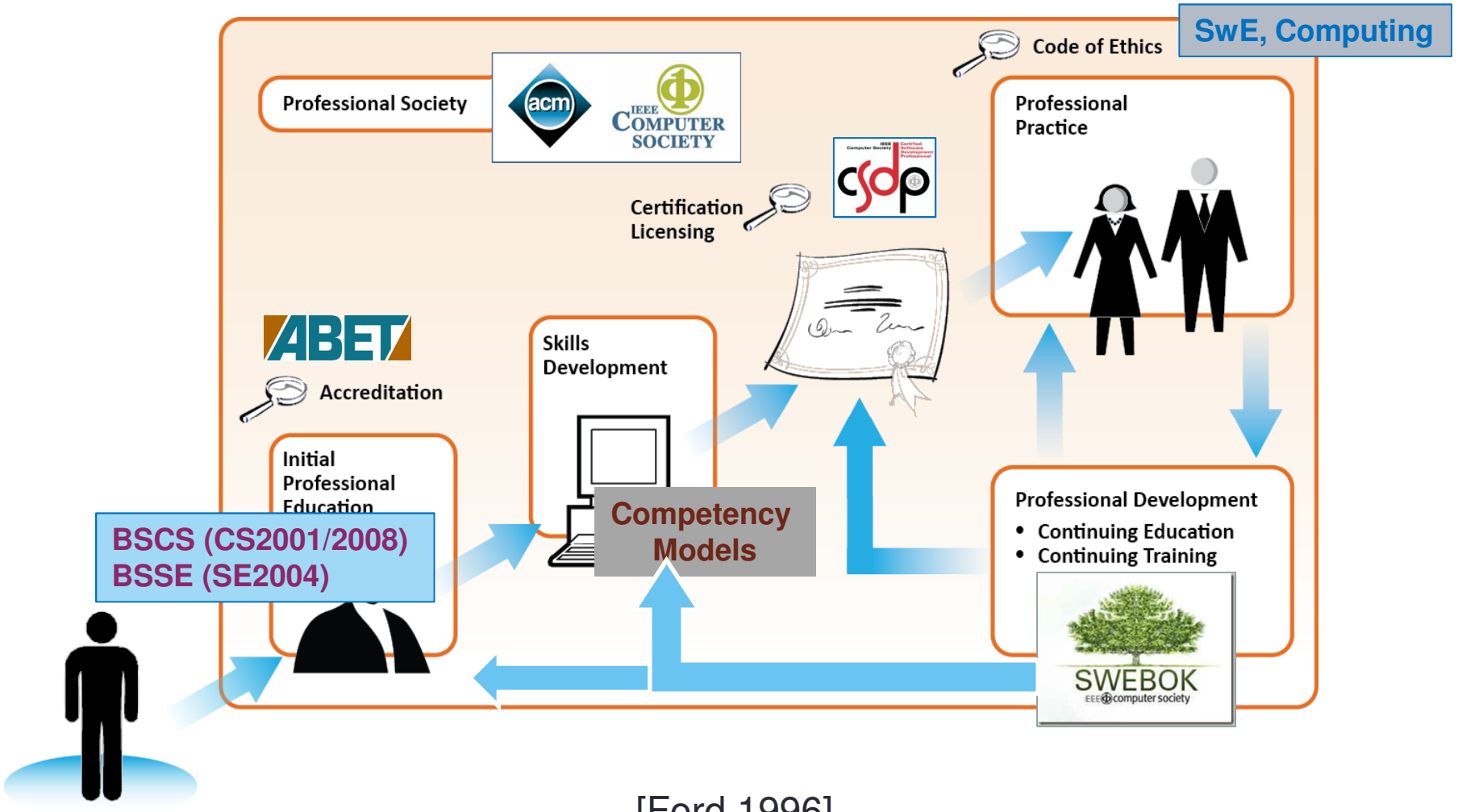
- Acquire and maintain the appropriate techinical knowledge and capability to work e
- Possess sufficient c problems in their discipline.
- Have the capability her professionals.
- Communicate effec loyer, and clients.
- Accept full respons
- Cooperate in efforts ublic concern related to their work
- Act fairly and avoid articularly public ones, concerning th
- Volunteer their prof and contribute to public education concerning their discipline



How are things 20 years later? (1)

- Software engineering practice as been significantly influenced by the following
  - advances in computing methods and technology,
  - modern software system complexity,
  - increased demand for software, and
  - the need to work across cultures and time zones have influenced significant change in the practice of software engineering.
- Employment of “software engineers” has improved.
  - Job Ranking (2012)
    - U.S. News ranks Software Developer as No. 7 job – based on employment opportunity, good salary, manageable work-life balance, job security
    - CareerCast ranks Software Engineer as No. 1 job - based on Environment, Income, Outlook, Stress and Physical Demands.
  - Job Availability (EngineerJobs.com - 5/10/13)
    - Lists 126,547 Software Engineering Jobs
    - Lists 17,673 Mechanical Engineering Jobs
  - BLS 2013 Job Outlook
    - Employment of software developers projected to grow 30 percent from 2010 to 2020, much faster than the average for all occupations.
    - The 2010 BLS Job Outlook changed Computer Software Engineer to Software Developer.
- There have been some significant advances in support of software engineering professional practice.

How does one become a Professional?



[Ford 1996]

ABET Student Outcomes (EAC)

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an **ability to design** a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on **multidisciplinary teams**
- (e) an ability to identify, formulate, and **solve engineering problems**
- (f) an understanding of **professional and ethical responsibility**
- (g) an ability to **communicate effectively**
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in **life-long learning**
- (j) a knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for **engineering practice**.

SwE Professional Practice Influences (1)

- CS 2013
  - "The education that undergraduates in Computer Science receive must adequately prepare them for the workforce in a more holistic way than simply conveying technical facts."
  - SwE and SE &Sys fundamentals - 97 contact hrs + CS - 294 contact hrs
  - Social Issues and Professional Practice - 16 contact hours
- SE 2004
  - "A key objective of any engineering program is to provide graduates with the tools necessary to begin the professional practice of engineering."
  - SwE knowledge – 322 contact hours + CS - 172 contact hours
  - Professional Practice (Group Dynamics/Psychology, SwE Communication Skills, Professionalism) – 35 contact hours
- SWEBOK 2013
  - Includes a chapter on Professional Practice
  - CSDP covers professional practice

SwE Professional Practice Influences (2)

- CSEET 2013 – lots of events related to professional practice
  - University Meets Industry: Calling in Real Stakeholders
  - Revisions to SE 2004
  - A Project Spine for Software Engineering Curricular Design
  - The Software Assurance Competency Model: A Roadmap to Enhance Individual Professional Capability
  - Software Engineering in CS 2013
- SIGCSE 2013
  - Computer Science Curriculum 2013: Social and Professional Recommendations
  - Gaps Between Industry Expectations and the Abilities of Graduates
  - A Case for Course Capstone Projects in CS1

Software Development Programs

- There are thousands of undergraduate software degree programs (CS, SwE, CE, IS, IT, etc.)
- ABET Accredited Programs
  - BSCS – 273 programs
  - BSSE – 26 programs
  - BSCE – 247 programs
  - BSIS – 47 programs
  - BSIT – 25 programs
- SwE ABET-EAC Program Criteria
  - *The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program.*
  - *The curriculum must prepare graduates to **analyze, design, verify, validate, implement, apply, and maintain software systems**; to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems; to **work in one or more significant application domains**; and to **manage the development of software systems**.*

How are things 20 years later? (2)

- 25 ABET Accredited BSCS programs reviewed in May 2013 (picked somewhat randomly)
- Program Educational Objectives
  - Most were a bit vague regarding professional practice – *"prepare students for a successful career in computer science"*
  - Many were not specially labeled or did not appear on the program website or in the school catalog; often appeared under link **ABET**.
  - Good example: http://www.msoe.edu/academics/academic_departments/eecs/bsse/objectives.shtml
- Software Engineering Courses
  - 14 had one required course
  - 7 offered software engineering only as an elective course
  - 4 offered no software engineering courses
- Senior-level Team Software Development Project
  - 17 out of 25 required a senior-level software project course

How are things 20 years later? (3)

- Although as educators we have made much progress, there are still serious problems in meeting Deming's Educational Challenge.
- A recent study [Radermacher 2013] of the gaps between CS graduates capabilities and industry expectations/needs showed the following "knowledge deficiencies":

Table 2. Most frequently identified knowledge deficiencies

Knowledge Deficiency	Occurences
Oral Communication	11
Teamwork	11
Project Communication	10
Problem Solving	10
Written Communication	9
Testing	9
Programming	8
Critical and Analytical Thinking	7
Design	6
Ethics	5
Configuration Management	5
Requirements	5
Programming Languages	5
User Interface Design	5

Success as an Engineer?

- A recent study [Passow 2012], using a survey of 4000 alumni of engineering programs, identified the four highest rated competencies (ABET outcomes):
  - “ability to function on a team”
  - “engineering problem-solving skills”
  - “ability to analyze and interpret data”
  - “written and oral communication skills”

Ability to Function on a Team

- “Students are not born with the project management, time management, conflict resolution, and communication skills required for high performance teamwork.” [Oaklley 2004]
- Bui
  - T
  - C
  - P
    re
    c
  - T
  - S
    - Have team solve problems on a fictitious team. (http://www.softwarecasestudy.org/)
  - Make sure teams have regular (weekly) deliverables and that the coach provides meaningful, timely feedback on teamwork.

g

Engineering Problem-Solving Skills

Faculty Interviews (with 25 colleagues)

How do you teach problem solving?

- “Give your students problems to solve.”
- “cover theory, then give lots of examples - different from the book”
- “ensure students have proper background and foundation for solving course problems”
- “start with very simple examples”
- “giving hints, seeding ideas”
- “fundamentals are necessary for problem solving”
- “provide students with an organized problem solving process”
- “help students to display there thinking”
- “show them how to do it, then have them try it out”
- “focus on things that make sense physically”
- “divide and conquer”
- “project-base teaching (with real customers)”
- “problem solving in one course supports problem solving in others”
- “stepwise refinement”
- “place responsibility for learning on students”
- “learn through doing”
- “teamwork enhances problem solving ability”
- “show how to set up problem”
- “cover theory, then work problems illustrating the theory”
- “use analogies to illustrate principles”
- “tell stories - use case studies”
- labs are the ultimate problem solving event”

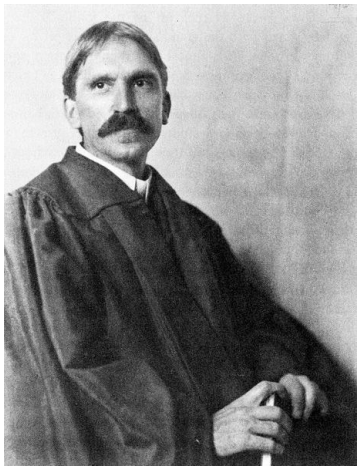
Other Thoughts on Problem Solving

It isn't that they can't see the solution. It's that they can't see the problem.

- Gilbert K. Chesterton

A problem well stated is a problem half solved.

- John Dewey



An inevitable consequence of the knowledge explosion is that tasks will be carried out with far more collaboration.

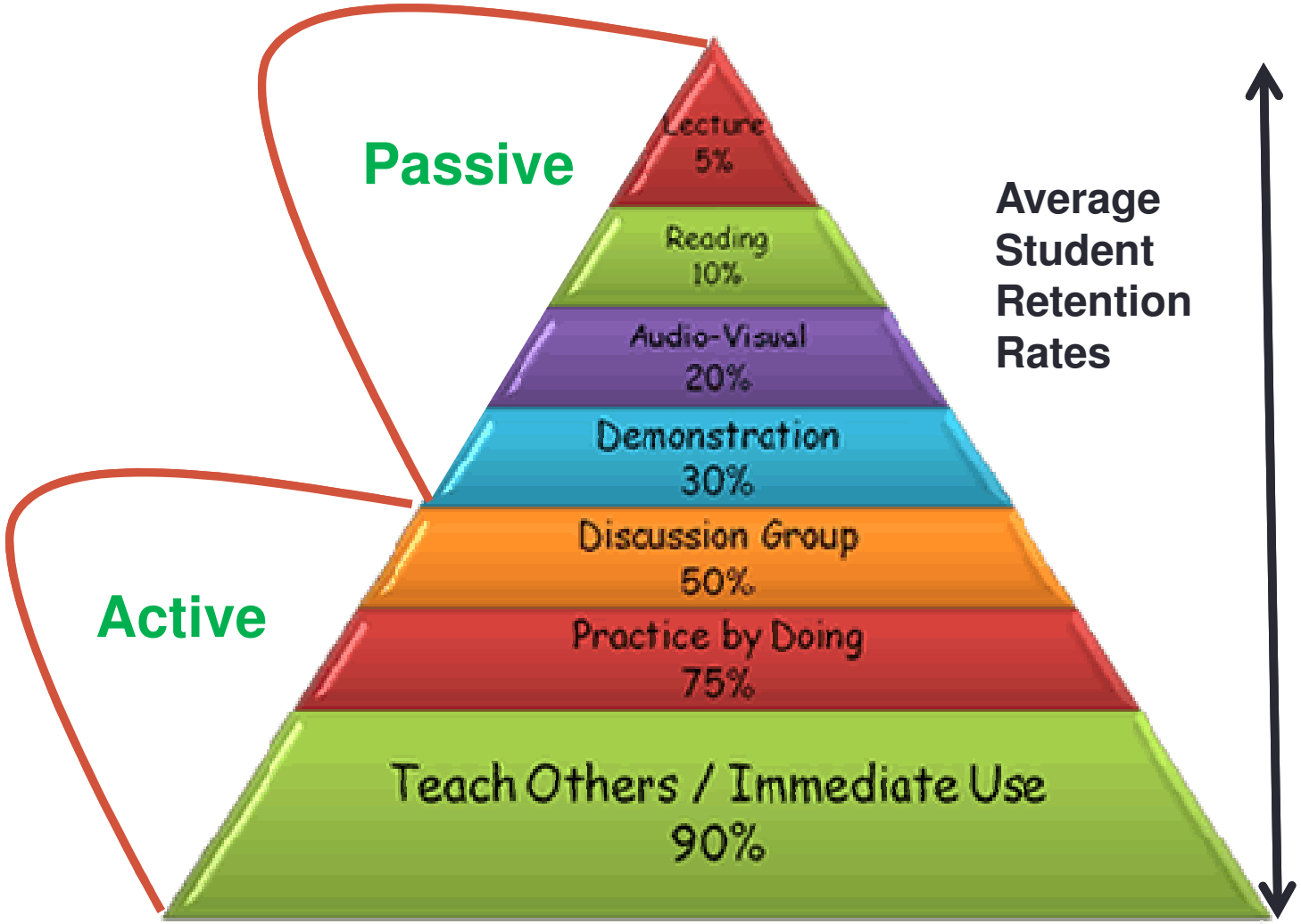
- Lawrence Summers

Understanding the Problem

Problem-Solving Strategies

- Collaboration – Teamwork
- Incremental/Iterative, Divide & Conquer, Top-Down, Bottom-Up
- Analogy and Reuse
- Defined Process [Decartes 1637, Deming 1986, Polya 1957]
  - Understand the Problem (determine need and scope)
  - Make a Plan (determine tasks, schedule, resources)
  - Carry Out Plan (specify, design/construct solution)
  - Check (verify/inspect/test solution)
  - Look Back (analyze process and product quality, in order to improve)

Educating Problem Solvers



[Dale 1969]

Active Learning

- In the last twenty years there has been considerable interest and research in applying active learning techniques and activities to improve student learning.
  - The teacher becomes a facilitator, guiding and coaching, rather than directing and lecturing.
  - Students learn by doing: class discussions, exercises, debates, study and analysis of case studies, and collaborative learning groups.
- "Tell me and I forget. Show me and I remember. Involve me and I understand."
  - An old Chinese proverb

A Life-Cycle Engineering Case Study

- The use of Case studies is an especially effective active learning technique for introducing realistic aspects of practice into a curriculum.
- The Digital Home Case Study [Salamah 2011] (http://www.softwarecasestudy.org/) is "life-cycle" case study designed to be used throughout a computing curriculum (CS1 to Senior Design), covering topics such as:
  - Requirements, Design, Construction, Testing, Maintenance
  - Project Planning, Risk Management, Configuration Management
  - Team Building
  - Quality Assurance
  - Process Management
  - Ethics and Professionalism

Ability to Analyze and Interpret Data (1)

- Analysis and interpretation of data is essential in an evolving, dynamic discipline, like software engineering.
- Data and its analysis can help us answer critical questions:
  - How does one evaluate and decide on best practices?
  - Is the latest popular " method du jour" a fad or practice that should be adopted?
- Unfortunately, proponents of a particular method, technique, or tool too often take rigid positions without looking at or seeking supporting data.
  - For several years, a key question at the SIGCSE Symposium was what is the best programming language for beginning programmers?
  - In a study of software development process models [Jones 2012], Caper Jones states **"selecting a software development method is more like joining a cult than a technical decision"**.

Ability to Analyze and Interpret Data (2)

- Although the students need to be exposed to research on software data collection and analysis, we also need to make sure our students do some of this on their own.
- Answering questions about their own work can help them to better see the importance of measurement and analysis:
  - How much effort is spent in various software development activities? E.g., the % of time in analysis & specification, design, construction testing, etc.
  - What is the quality of your work? E.g., number and type of defects found, defect removal effectiveness, cost of quality, etc.
  - How well are quality attributes achieved? E.g., usability, performance, maintainability, etc.

Written and Oral Communication Skills

- We should not depend solely on “communication” courses.
- Technical communication(written and oral) should be a prominent part of the curriculum, especially in project courses.
- We often fail to recognize the importance of general education in fostering communication skills (history, literature, philosophy, psychology, physical sciences).
- SWEBOK 2013 will have chapter on Professional Practice, which includes a section on communication.
- Unfortunately, we often overlook the importance of “reading” and “listening” skills in effective communication.
  - Reviews and inspections require careful and focused reading.
  - [Radermacher 2013] -- “the biggest discrepancy between employer expectations and student ability was their ability to listen”

Professional Practice Teaching Challenges

- Teaching in an Academic Setting
  - Curriculum is divided into chunks of academic terms and courses. Student effort is divided between seemingly unrelated and incongruent curriculum units.
  - Education is typically delivered by lecture and individual homework assignments. Course grades are generally assigned on the basis of individual work;
- Many, if not most, faculty are not properly prepared to teach professional practice.
  - Lack experience in professional practice in their discipline.
  - Focus their research and teaching on a narrow subfield of the discipline.
  - Are not motivated to do engage in collaboration (such as team teaching).
  - Lack preparation in teaching techniques that best serve preparing students for professional practice.
- A survey of teaching in engineering departments shows that 74 % of computer science and software engineering faculty were aware of innovative teaching techniques (e.g., student-active pedagogies, first year design projects, and artifact dissection), but less than 40% of these faculty members use such techniques – the lowest percentages of the faculty disciplines surveyed [Borrego 2010].

Ideas for Meeting Deming's Challenge

- Curriculum goals need to be the central driving element in meeting the challenge.
- Faculty need a better understanding of professional practice.
  - Professional Experience - Faculty Internships
  - Industry Visits and Tours
  - Interview Professionals
  - Industrial Advisory Boards
  - Industry Guest Lecturers
  - Applied research – helping industry to solve problems
  - Study of Competency Models
  - Attend CSEET 2014
- Faculty need to Embrace "Active Learning"
  - Spread project work and team activities throughout the curriculum.
  - Use case studies and other student-centered learning exercises
  - "Coach" rather than "Lecture"
  - Emphasize a Students' responsibility for their learning.
  - Learn more about how to build effective teams.

Questions/Comments?

Sources

- [Borrego 2010] Borrego, M., J. Froyda, and T. Siminhall, Diffusion of Engineering Education Innovations: A Survey of Awareness and Adoption Rates in U.S. Engineering Departments, *Journal of Engineering Education*, (July 2010), 185-207.
- [Dale 1969] Dale, E , *Audiovisual Methods in Teaching*, 3rd edition, Holt, Rinehart and Winston, 1969.
- [Decartes 1637] Decartes, Rene', (translated by Norman K. Smith) Discourse on Method, *Descartes Philosophical Writings*, The Modern Library, pp 90-159, 1958.
- [Deming 1986] Deming, W. Edwards, *Out of the Crisis*, MIT Press, 1986.
- [Denning 1992] Denning, P.J., Educating a New Engineer, *Communications of the ACM*, pp 83-97, December 1992.
- [Ford 1996] Ford, G. and Gibbs, N. E., *A Mature Profession of Software Engineering*, CMU/SEI-96-TR-004, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1996.
- [Jones 2012] Jones, Caper, *Evaluating Ten Software Development Methodologies*, Version 1.4, Capers Jones & Associates , January 2012 (http://www.ppi-int.com/newsletter/SyEN-040.php#article)
- [Keller 1998] Keller, Robert & Concannon, Thomas, Teaching Problem Solving, *for your consideration…*, Center for Teaching and Learning, University of North Carolina at Chapel Hill, No. 20, June 1998.
- [Polya 1957] Polya, G., *How to Solve It: A New Aspect of Mathematical Method*, 2nd Ed., Princeton University Press, 1957.
- [Radermacher 2013] Radermacher, A. and Walia, G., Gaps Between Industry Expectations and the Abilities of Graduates: Systematic Literature Review Findings, Proceedings of SIGCSE 2013, March 2013.
- [Salamah 2011] Salamah, S. Towhidnejad, M. and Hilburn, T., Developing Case Modules for Teaching Software Engineering and Computer Science Concepts, *Proceedings of 2011 Frontiers in Education Conference*, October 2011.
- [Summers 2012] Summers, L., What You (Really) Need to Know, *New York Times*, January 20, 2012.
- [Tomey 2003] Tomey, A., Learning with Cases, *Journal Of Continuing Education In Nursing*, Vol 34, No 1, January/February 2003.
- [Yin 2009] R. K. Yin. Case Study Research: Design and Methods (Applied Social Research Methods). Sage Publications, fourth edition. edition, 2009.