This talk explains how the confluence of cloud computing and Massive OpenOnline Courses (MOOCs) have allowed us greatly improve both the effectiveness and the reach of UC Berkeley's undergraduate software engineering course.

Myths About MOOCs and Software Engineering Education

David Patterson is the Pardee Professor of Computer Science at UC Berkeley, which he joined after graduating from UCLA in 1977. Dave’s research style is to identify critical questions for the IT industry and gather inter-disciplinary groups of faculty and graduate students to answer them. Patterson and his teammates have won the following awards: the C & C Prize, the IEEE von Neumann Medal, the IEEE Johnson Storage Award, the SIGMOD Test of Time award, the ACM-IEEE Eckert-Mauchly Award, and the Katayanagi Prize. He was also elected to both AAAS societies, the National Academy of Engineering, the National Academy of Sciences, the Silicon Valley Engineering Hall of Fame, and to be a Fellow of the Computer History Museum. The full list includes about 35 awards for research, teaching, and service. In his spare time he coauthored six books, including two with John Hennessy, who is President of Stanford University. Patterson also served as Chair of the Computer Science Division at UC Berkeley, Chair of the Computing Research Association, and President of ACM.

For more information, please contact the Chair’s Office at (858) 822-1516 or cse-cc@eng.ucsd.edu
Title: Myths about MOOCs and Software Engineering Education

Abstract: This talk explains how the confluence of cloud computing and Massive Open Online Courses (MOOCs) have allowed us greatly improve both the effectiveness and the reach of UC Berkeley's undergraduate software engineering course.

The first part of the talk is motivated by Industry's long-standing complaint that academia ignores vital software topics, leaving students unprepared upon graduation. Traditional approaches to software development are neither supported by tools that students could readily use, nor appropriate for projects whose scope matched a college course. Hence, instructors traditionally lecture about software engineering topics, while students continue to build software more or less the way they always had, in practice relegating software engineering to little more than a project course. This sad but stable state of affairs is frustrating to instructors, boring to students, and disappointing to industry.

Happily, cloud computing and the shift in the software industry towards software as a service has led to highly-productive tools and techniques that are a much better match to the classroom than earlier software development methods. That is, not only has the future of software been revolutionized, it has changed in a way that makes it easier to teach. UC Berkeley’s revised Software Engineering course leverages this productivity to allow students to both enhance a legacy application and to develop a new app that matches requirements of non-technical customers. By experiencing the whole software life cycle repeatedly within a single college course, students actually use the skills that industry has long encouraged and learn to appreciate them. The course is now rewarding for faculty, popular with students, and praised by industry.

The second part of the talk is about our experience using MOOCs to teach Software Engineering. While the media's spotlight on MOOCs continues unabated, a recent opinion piece expresses grave concerns about their role ("Will MOOCs Destroy Academia?", Moshe Vardi, CACM 55(11), Nov. 2012). I will try to bust a few MOOC myths by presenting provocative, if anecdotal, evidence that appropriate use of MOOC technology can improve on-campus pedagogy, increase student throughput while raising course quality, and even reinvigorate faculty teaching. I'll also explain the role of MOOCs in enabling half-dozen universities to replicate and build upon our work via Small Private Online Courses (SPOCs) from EdX and our electronic textbook.

I conclude that the 21st century textbook may prove to be a hybrid of SPOCs and Ebooks.

Biography: David Patterson is the Pardee Professor of Computer Science at the University of California at Berkeley, which he joined after graduating from UCLA in 1977.

Dave's research style is to identify critical questions for the IT industry and gather inter-disciplinary groups of faculty and graduate students to answer them. The answer is typically embodied in demonstration systems, and these demonstration systems are later mirrored in commercial products. In addition to research impact, these projects train leaders of our field. The best known projects were
Reduced Instruction Set Computers (RISC), Redundant Array of Inexpensive Disks (RAID), and Networks of Workstations (NOW), each of which helped lead to billion dollar industries.

A measure of the success of projects is the list of awards won by Patterson and as his teammates: the C & C Prize, the IEEE von Neumann Medal, the IEEE Johnson Storage Award, the SIGMOD Test of Time award, the ACM-IEEE Eckert-Mauchly Award, and the Katayanagi Prize. He was also elected to both AAAS societies, the National Academy of Engineering, the National Academy of Sciences, the Silicon Valley Engineering Hall of Fame, and to be a Fellow of the Computer History Museum. The full list includes about 35 awards for research, teaching, and service.

In his spare time he coauthored six books, including two with John Hennessy, who is President of Stanford University. Patterson also served as Chair of the Computer Science Division at UC Berkeley, Chair of the Computing Research Association, and President of ACM.