David Patterson

Interview
An Interview with Stanford University President John Hennessy

Stanford University President John Hennessy discusses his academic and industry experiences in Silicon Valley with UC Berkeley CS Professor David Patterson.

JOHN HENNESSY JOINED Stanford in 1977 right after receiving his Ph.D. from the State University of New York at Stony Brook. He soon became a leader of Reduced Instruction Set Computers. This research led to the founding of MIPS Computer Systems, which was later acquired for $320 million. There are still nearly a billion MIPS processors shipped annually, 30 years after the company was founded.

Hennessy returned to Stanford to do foundational research in large-scale shared memory multiprocessors. In his spare time, he co-authored two textbooks on computer architecture, which have been continuously revised and are still popular 25 years later. This record led to numerous honors, including ACM Fellow, election to both the National Academy of Engineering and the National Academy of Sciences.

Not resting on his research and teaching laurels, he quickly moved up the academic administrative ladder, going from the CS department chair to Engineering college dean to provost and finally to president in just seven years. He is Stanford’s tenth president, its first from engineering, and he has governed it for an eighth of its existence. Since 2000, he doubled Stanford’s endowment, including a record $6.2 billion for a single campaign. He used those funds to launch many initiatives—which often cross departmental lines—along with new buildings to house them. Undergraduate applications also doubled, for the first time making Stanford even more selective than Harvard.

This interview occurred in the Stanford President’s office in July 2015, shortly after he announced that he would step down as president in 2016—Stanford’s 125th anniversary. The interviewer is David Patterson, who is a UC Berkeley CS professor and co-author of Hennessy’s two textbooks.

Ending of Moore’s Law

DAVID PATTERSON: You received your bachelor’s degree in electrical engineering 40 years ago. Amazing things have happened since then in the information technology. What do you think about the next 40 years?

JOHN HENNESSY: You are absolutely right it has been transformative. We
Trying to retreat from technology to preserve old jobs didn’t work in the Luddite era, and it’s not going to work today.

When I tell people that Moore’s Law is ending, they think it means the technology is going to stop improving. We know technology advancement is not ending, but it’s going to slow down. What do you think the impact will be?

Hardware has become so cheap that people don’t think about throwing away their phone and getting a new one because it’s a little bit better. We’ll see those things begin to slow down. There hasn’t been doubling for quite some time now, so we’re already kind of in that sunset period. The challenge is if we continue to be inventive about the way we use information technology, how are we going to deliver the hardware that enables people to use it cost effectively?

Do you mean design?

Yes. How are we going to continue to make it cheaper? We still don’t have our hands completely around the issue of parallelism. We switched to multicore, but we have not made it as useful as if we had just made single-threaded processors faster. We have a lot of work to do.

Since transistors are not getting much better, some think special-purpose processors are our only path forward.

Clearly, there has been this dance back and forth over time between general purpose and special purpose. For well-defined applications, special purpose can yield lots of performance, particularly when signal processing intensive. The question will always be how general those things can be and how they compete against general-purpose machines. With a well-defined task, you can build hardware that does it very well.

The flip side of that is we’re not done with software and new algorithms, and those are going to be much harder to put onto special-purpose pieces of hardware before we understand them.

Can we match the software to the hardware?

That’s the hard problem. And it’s one that we have made relatively little progress on. It requires probably both advances in compiler technology and also in how we think about programming those machines.

Impact of CS on Jobs

There is increasing concern about advances in information technology and what it is going to do to jobs. Changes were generational in the past, so you would do a job that would go away, but your children could learn another job. However, given the speed of change of information technology, jobs change within a single career.

A looming example is self-driving cars. It promises huge societal benefits—fewer deaths and injuries, helping the elderly or the disabled—but many jobs would go away: taxi cab drivers, car repair shops, car insurance agents. Should CS, academia, or society take action, or should we just let market forces run their course?

Trying to retreat from technology to preserve old jobs didn’t work back in the Luddite era, and it’s not going to work today. You’ve got to figure out how you’re going to retrain and redeploy people into new kinds of career opportunities. The interesting thing about self-driving cars is not only could we achieve some economic efficiency probably, but also we can achieve dramatic improvement in quality of life for everybody that’s involved in it.

But I think it is important to think about technology. It is not that technology can’t be abused; it can obviously be used for doing bad things as well as good things. We will face this problem with machine learning as AI gets better and better. The right solution is to figure out how to use it effectively and how to create value for society.

That will require us to retrain people and put them into new professions. Thomas Friedman once said, “The only way to keep up with the change in the world is to run faster. You can’t run slower and try and retard the progress.”

So we can’t stop the change. Is there something that those of us that are inventing the technology that is changing the world can do to soften the impact of job displacement, or is that outside of our skill set?

It’s hard to figure out exactly what role we have. Clearly, making education cheaper because in the end you’re going to retrain people through education. For retraining functions, online education is the right way to go. So there’s an opportunity to really try to get people involved in new things.

I can see why people like interviewing you; that’s a good idea. If we’re educators and there’s going to be this massive displacement of jobs, we can either relive the French revolution or we can help improve retraining.

Whither MOOCs

There’s been this theme over your career about computing and education. You had faculty here a few years ago that said massive open online courses (MOOCs) were going to revolutionize campuses, offering Stanford degrees for a few dollars. That didn’t happen. What do you think is going on with MOOCs?

MOOCs have both advantages and
disadvantages. They're certainly not the end-all and be-all answer. Their advantage, namely cost of production, comes from one course serving lots of students with very wide backgrounds and very little requirements in terms of prerequisites, but that's their disadvantage, too. It means that you've got a group of students who are very widely distributed in terms of ability.

For some of them, the course will be going too slow, for some it will be going way too fast, for some it will be too challenging, too easy. That's a really hard problem to solve. I think technology clearly has a role to play here. And if you said: well, I'm going to use it to teach moderate-sized courses of screened students, where they've had certain prerequisites, then I think there's a great role there and it's a way to probably get some increase in scale in our institutions.

There's another key role, which I think is critical here. In much of the developing world, there is just no access at all to higher education. You won't deliver access unless you can do it extremely inexpensively—there MOOCs have a real role to play. I think we have already seen the role that MOOCs can play in educating the educators in these institutions, as well as providing students without any other alternative with a route to education. We have to improve the way we do things in higher education. Clearly, we've got to become more efficient over time, but to try to make a quantum leap to MOOCs is probably not the right way to do it.

My colleague Armando Fox and I were early MOOC adopters. Our attraction was the large international audience and solving the continuing education problem.

Continuing education; 100% this is going to go online very quickly, whether it goes to MOOCs or large classes where students have similar backgrounds, but it's the only way to really solve two problems, namely people who are out there working are simply too busy to come to an institution. The time shifting that occurs and balancing people's schedules, so I think online will happen. I wouldn't be surprised to see professional master's degrees become a hybrid, partly online, partly experiential. In a field like ours, you do want some interaction, teamwork, experimental work, this kind of thing.

Armando Fox says it is currently like early days of CS where everything sucked. We eventually made it better. This era is the early version of MOOC software, and things might get better.

The real Golden Fleece is to actually get a system that does adaptive learning, and really responds to how are people learning. You did this little quiz question, now what do you do? The big problem in education is, in some sense, motivation. How do you get students to really engage? Because simply looking at whether it's an instructor sitting in a classroom passively or sitting in front of a screen passively is not a learning experience. There is some good data out there. The data around so-called interactive active learning models does work because it engages students more, and that is what we've got to focus on. We've got to experiment, try things, and the academy is sometimes a little reluctant to experiment.

Critiques of His Presidency

As far as I can tell in your amazing 15-year record, the biggest criticism of your tenure—which appeared, for some reason, in The New Yorker—is that students leave to do startups and that you served on several corporate boards. Let me do these one at a time. There can't be very many undergraduate students leaving. Essentially none.

Are graduate students leaving?

Essentially none. This was driven by one story about one company, where the student happened to be my advisee, and I told him to finish your degree and talk to your parents about what you do. He did finish his degree. All his co-founders, all his startup team finished their degrees.

What's the story there?

We have a program through StartX, where students will get enough money to eat beans for the summer and try out their idea and see if it works. Most of them conclude: “Well, it's worth continuing working on, but it's not the hottest thing going.” For undergraduates, we try to emphasize the value of their undergraduate degree and what that's going to mean in their long-term future, not their short-term future, and that's worked just fine.

It's a whole different situation with graduate students. If you are Larry Page and Sergey Brin or are Jerry Yang and David Filo, and you can create Google or Yahoo, you ought to go do that.

The second issue was where there were doubters even on your campus about whether a president should be serving on boards of corporations. If I was in your shoes, I'd think I'm going to be running this big organization, so if I can serve on boards and see how these companies deal with challenging problems, maybe I'll get some ideas.

Yes.

With the economic turmoil in 2007, you did cuts in advance of the big downturn at Stanford, whereas most other college presidents reacted after it happened.

Certainly from my experience from being on boards, I saw what was happening. I saw how much the endowment was going to be impacted. I have learned having gone through layoffs at MIPS that the best thing to do is to get through it as quickly as possible and reset the budget and reset the new normal and then move forward. That's what we tried to do and we recovered very fast. Things were over in a year or 18 months for Stanford. We began then a positive course forward.

Anything else?

One of the most important things I think I've imported from boards is to focus on leadership training and leadership preparation.

We don't do a good job of this in the university. We have an opening, and we find that we don't have anybody who has the right kind of background and experi-
Grading on the curve, it’s pretty good.

What does university leadership training look like?

The big one we do is the leadership academy, that’s our kind of premier program. It’s taught by Chuck Holloway from the business school and John Morgridge, the former CEO of Cisco. It’s everything from: How do you think about solving problems? How do you think about issues of diversity and equity in the university? Looking at various case studies, how do universities deal with public affairs crisis, but then also some personal growth, whether it’s communications coaching or it’s coaching on how do you deal with difficult problems, getting reviews, giving people feedback, those are all things you have to learn along the way. Learning them and having some exposure to them certainly helps people when they’re in a position where they have to do it.

The Magic of Silicon Valley

Both of us came to the San Francisco Bay area in 1976–1977. I am just amazed at the success of this region. It has arguably the number-one private university and arguably the number-one public university. You’ve got the leading information and technology center in the world, one of the best wine regions, and even our professional sports teams do well.

I know we have nice weather and we’re near the Pacific Ocean and the Sierras, but it’s got to be more than climate and geography. Is it that our universities are attracting and training great people, and they stay there, the lure of Silicon Valley, or ...?

A couple of other things helped. This is a place that is very welcoming to people from around the world. No matter where you come from, you can probably find people who are similar to you in the Valley. It’s not a perfect meritocracy, it’s a ways from it—but it’s a pretty good meritocracy.

Grading on the curve, it’s pretty good.

John Hennessy and David Patterson circa 1991 with their textbook *Computer Architecture: A Quantitative Approach*, which was published the year before and is now in its fifth edition.

Yes. And the curve is very uneven, and it’s not always equitable, and certainly there are things, biases and things that come in, but it’s pretty good overall. It’s also going back to the days of Hewlett and Packard; it tends to have organizations that are relatively flat. They tend to have open management styles.

That’s the tradition?

The tradition of Hewlett and Packard and then Intel after that. It’s a tradition in the Valley that everybody’s got a role to play and everybody can make a contribution, even in a large company. You and I both came when if you wanted to talk to the movers and shakers in the computer industry, we had to get on a plane and fly to New York or fly to Boston.

Now it’s the other way around. Creative destruction is embraced in the Valley, and that’s why things happen; the old makes way for the new. In creating that capability, management and leadership talent as much as engineering talent becomes key. It’s great to have a team of young engineers, but if you don’t have anybody on that team that’s ever delivered a product, they’re challenged to figure out what the issues are and how do you make that work.

CS Popularity on Campus

We’ve seen an explosion in the popularity of CS everywhere; CS is now the biggest major on your campus. What is going to happen? CS faculty size is going to have to grow in response to demand, or since it’s a zero-sum game, only slow changes?

We’re going to struggle to meet the demand. All institutions are facing this because CS graduates have lots of good opportunities, both within the academy as well as outside it. We’re also going to see other fields begin to change, but change takes a long time. Universities change at a very slow pace getting new faculty.

However, I think we can already see the rise of quantitative empirical analytical approaches, for example take a discipline like political science. The action is in doing big data analysis of elections and surveys. It requires a different kind of expertise and background. We will slowly see the political science department starting to focus on training people. The next generation of young faculty they’re bringing in—young meaning less than 50 years old—does a lot of this kind of work, right?

Sounds young to me.

I think we’ll see a shift that we’ll still have to teach them some basic CS, but maybe instead of being a major, they’ll be a minor, and they’ll do their work in this other discipline.

If computer scientists do not do the teaching, I worry about the quality of the courses over time. I think to do that you probably need more professors. Another way of asking this: Do you think schools of CS are going to get more popular on campuses, or will CS stay in college of
engineering?
I think it could stay. It’s a question of how big can it be. They are now one of the biggest departments, but compared to a medical school, they are small. Maybe we need to rethink how that plays. CS can grow and merge over time.

Publishing: Limiting the LPU
There is a lot of concern in CS about publishing. When you see how many dozen papers new Ph.D.’s have, it looks like lots of LPU (least publishable unit), where quantity trumps quality. One proposed solution is to limit the number of papers that count on job applications or for promotion. Are we the only field with the problem? Is the proposed solution plausible?

Other fields do a lot of publication. Look at the biological sciences and look at the list of publications they have. They would even have more than many computer scientists. I do worry that in this drive to publish and the use of conferences that we have become a little incremental. It’s easy enough in the academic setting to say: tell me about the five or six things that you’ve done in preparation for tenure? What are the most important five or six papers? Because the reality is, nobody is going to look much deeper than that, if they look carefully. We need to move away from just counting the number of publications and saying: okay, that’s enough publications. What has the person really done? In the end, it’s about impact. What impact did they have, either on an industry or on other researchers; because that’s what you really care about when you’re trying to evaluate the research work of a faculty member.

So then plausibly having a policy that limits to what your five best works or something like that?
Yes. I don’t think it would be a big issue.

Computer Science and X
From your view at the top of one of the great universities, what promising new fields should CS interact with?
One is really trying to crack the genome. What is this genome code for and how does it match against human disease and other kinds of health problems? We have made progress, but there is a lot more to do.

We’re seeing the era of big data creating interesting changes in the social sciences. People have recently studied life outcomes of people looking at the IRS database to see what determines what your economic prosperity is going to be over your life. Where you were born? Where you were educated? What kind of family you lived in? What kind of community you lived in? What kind of schools of thought? Trying to do that experiment as a social science experiment, where I take let’s say two schools, I run method A, I run one method B, is very hard and it takes a long time.

On the other hand, if you can find two examples of schools that have been run by two different methods and look at the data historically, and look at what the outcomes are like, you can get insights that are impossible to get any other way. We’re going to see a rise of the influence of big data in the social sciences, and they’re going to become more quantitative in how we formulate policy.

---

Rising College Costs
Changing topics again, you’re concerned about the rising cost of college tuition, as going faster than inflation just isn’t sustainable. What are your thoughts about getting college tuition back under the rate of inflation?

The first thing to say is that college is still a good investment. If you look at the data, it is overwhelmingly clear that college is a good investment. As I told a group of parents recently, you’re better off investing in your kid’s education than you are investing in the stock market—the return is better.

In return to the parent?
In return to the family. Well, if you care about the economic outcome of your kids. What we’ve got to do is figure out a way to kind of balance costs better. We’re going to have to find ways to figure out how to keep our costs under control and bend the cost curve a little bit. It doesn’t take much, but it takes a little so that the growth is something that families can deal with.

Note this is driven as much by wage stagnation in the U.S. If salaries and wages in the U.S. were still going up at faster than inflation, which was traditional, it would remain affordable, but they’re not, and so we’ve got this dilemma that we’re going to have to solve.

Hennessy’s Past and Future
Let’s go back to high school. Were you class president or valedictorian?
I was kind of a science nerd. My big science fair project was building a tic-tac-toe machine with my friend Steve Angle out of surplus relays, because at that time real integrated circuits were too expensive. It had green and red lights for machine and person. Lots of people don’t realize tic-tac-toe is a very simple game. When I brought it to see people, and likeable: is there anything else that you need to be a successful administrator?

You need to listen.

Be a good listener.
You need to keep in touch with people. You need to hear what they’re thinking. The people who really lead the university are the faculty in the end. Those of us that are in administrative roles are enablers, and we look for opportunities that can come by making connections across pieces of the institution. In order to do that job well, you really have to know what the faculty are thinking and what they are thinking about where we should go.

I spend a lot of time just interacting with faculty and listening to what they’re thinking about, where they see an opportunity. The other great thing is that students are terrific, they’re really great.

You’ve been spectacular at everything you’ve done in your professional career:
teacher, inventor, researcher, book author, entrepreneur, and administrator. You’ve been a role model to a lot of us. It made me wonder if I had challenged myself enough. (In fact, that’s why I ran for ACM president.) This may be a strange question for Stanford’s president, but have you ever wondered whether you have challenged yourself enough?

It’s always a question: Are you willing to take on any new challenges? Are you worried about doing it? I don’t think I’d be particularly good in government because I just don’t quite have the patience. Being an academic leader requires some amount of patience, but being in government requires a lot more patience.

It takes even longer.
Having talked to people who do those jobs, I mean, whether it’s the President or people who serve in Congress, that’s really hard work, and it requires a real depth of patience and resilience. That’s not my style; I mean I come from Silicon Valley. I’m a change person—and I wouldn’t be a good caretaker either. If what the university needed was a caretaker president, I would have been the wrong person to pick because I wasn’t going to find that job rewarding.

Let me wrap up. You have been called the most outstanding president of your generation. I see you as an intellectual descendant of two great California educators, Clark Kerr and Fred Terman. You have been a shining example for CS; all computer scientists benefit from the reflected glory from career. And you’ve been a wonderful colleague, co-author, and friend. You even disproved Leo Durocher’s saying, by showing that nice guys can finish first. A grateful field thanks you for what you’ve accomplished and the style with which you’ve done it. 

David Patterson
(pattrsn@cs.berkeley.edu) is the E.H. and M.E. Pardee Chair of Computer Science at UC Berkeley and a past president of ACM.

Copyright held by author.

For the full-length video, please visit https://vimeo.com/14345543

Watch the authors discuss their work in this exclusive Communications video. http://cacm.acm.org/videos/an-interview-with-stanford-university-president-john-hennessy