But there wasn't a lot of attention being paid to helping us students to produce beautiful things. In calculus, for instance, all the things they show you are 100 years old and have gotten a whole lot of polishing between then and now. And I was hoping for that in my math courses and I didn't get it.

When I switched to computer science I thought maybe there would be a better chance of that. Sure enough, it was still the case that the instructor would put up a beautiful program and say "look, isn't it beautiful?" and we had to agree.

A lot that was missing there in both those cases was not the end products of the theorem that's getting proved, or whatever, but the internal decisions that you made one way or another or whatever to move your way from the "to prove" statement to the completed proof.

And so that's where case studies came from.

What we're trying to do in a case study: typically, in the case studies we've put together, there's a problem statement, and there's one or maybe more solutions, and there's as much of the internal intermediate decisions that we made one way or another, and there's a narrative of those to make them accessible to the student.
We've had a lot of success with them in a variety of contexts. I think some of them we've discussed earlier in the interview.

One last thing about case studies is that I love to write them. With Marcia Linn, with whom I co-authored two books of case studies, the way we'd interact was that I'd put together a draft, and I'd hand it off to her, and she would make a lot of red marks on there, most of which that said

"Why?
Why did you do it this way rather than that way?
Why did you do that here rather than somewhere earlier or later?"

I relished getting that advice. It may sound very Californian, but I really get excited learning something or knowing something that I didn't know that I knew. Seems like very Californian somehow.