



# Why We Need Methods Courses


Dr. Chris Stephenson


CE21 PI Meeting






## A Community of Capable Teachers

- 26,000 teachers in the US who define themselves as CS or programming teachers.
  - We know nothing about what CS teachers know about computer science or about teaching
  - The current generation of teachers are retiring and when they leave, CS dies in their schools
  - Educational politics and industry make it more difficult for CS teachers to be hired and retained
- 




Without some way of determining that CS teachers know what they need to know, we will never be perceived as a legitimate academic discipline K-12.





## Existing Models


- We have looked at three CS methods courses
    - Purdue University
    - University of Windsor
    - Milwaukee School District
  - The primary differences relate to overall focus:
    - Programming
    - Applications
    - Project-based learning
- 

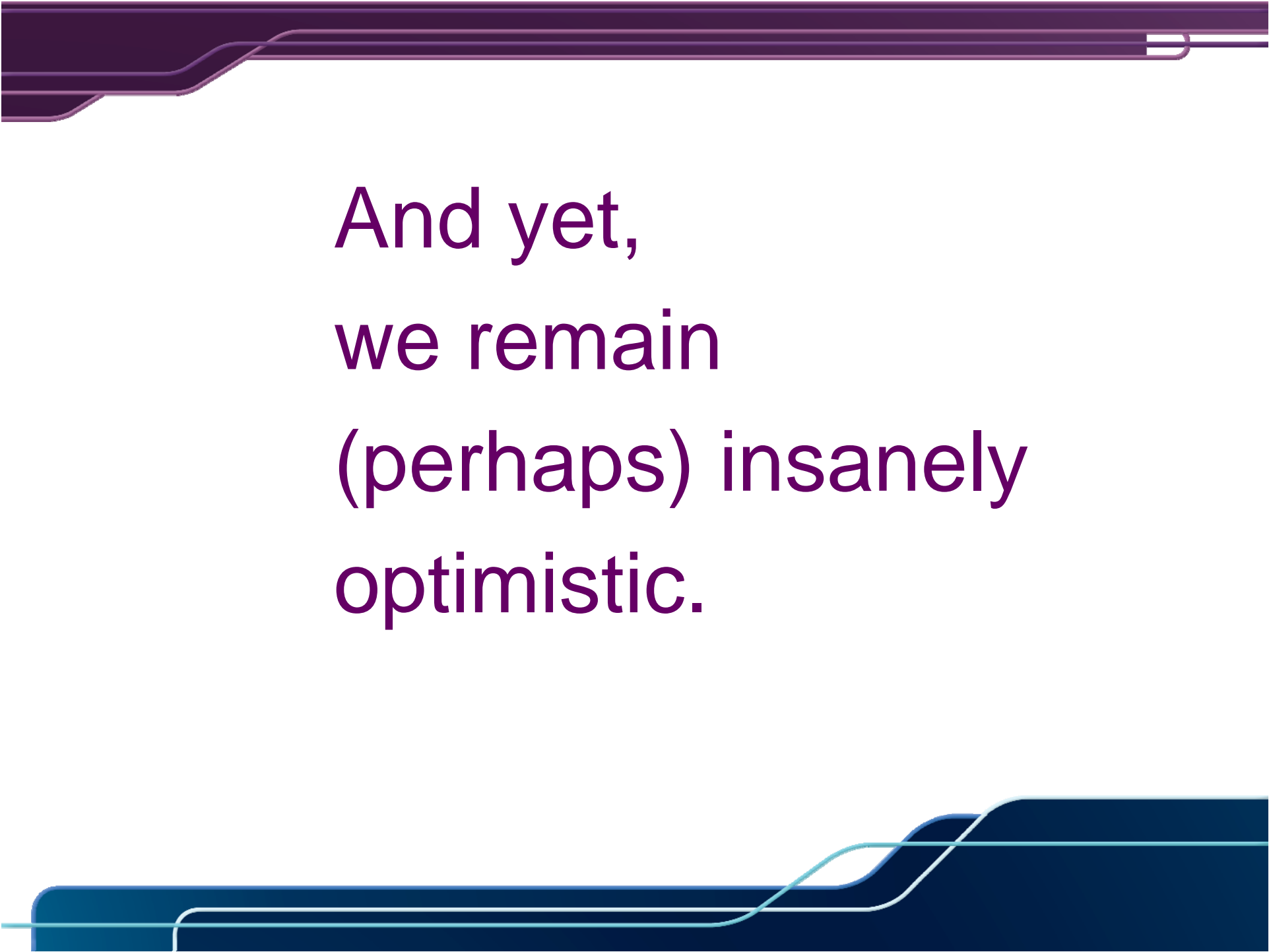
## Teacher Certification is a Mess

- Most states do not have CS certification so teacher preparation programs don't care
- Where programs exist, the requirements are either ridiculously complex, completely irrelevant, or impossible to meet
- Current certification programs (National Board Certification) are expensive and many states will not recognize them.
- We have good models for certification but states have other priorities right now.



## Tools, Teachers, Time, and Money

- The CS obsession with tools leads to tool-focused preservice and inservice learning
  - The teacher audience for CS methods courses has diverse and diverging needs
  - We are operating at a research deficit
  - We are operating at a policy deficit
  - We are operating at funding deficit
- 



And yet,  
we remain  
(perhaps) insanelly  
optimistic.



## Contact Information

CSTA website: [csta.acm.org](http://csta.acm.org)

Chris Stephenson  
Executive Director, CSTA  
Phone: 1-541-687-1840  
Fax: 1-541-687-1840  
[cstephenson@csta.acm.org](mailto:cstephenson@csta.acm.org)



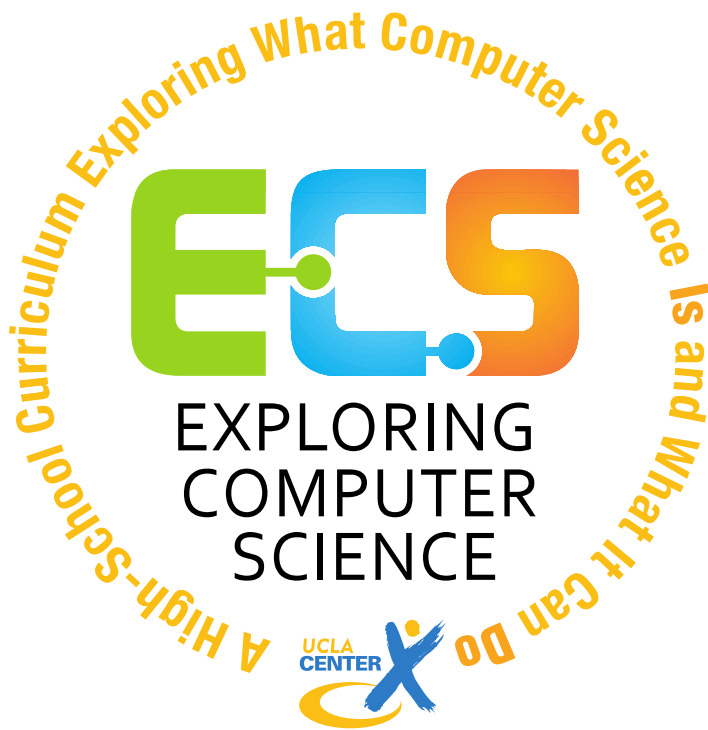
# Developing a Computer Science Teaching Methodology Course



February 3, 2012

Joanna Goode, University of Oregon

# CONTEXT



# PROCESS



- Develop preliminary syllabus based on educational research in computer science education and teacher education
- Feedback from CS Education Community
- Meeting of Methodology Faculty at SIGCSE
- Revising syllabus based on feedback
- Pilot course in the Spring with 22 mathematics and science pre-service teachers at UCLA
- Revise syllabus based on feedback

# OVERVIEW OF COURSE



This course for prospective secondary STEM teachers explores computer science as a discipline that encourages inquiry, creativity, and collaboration. The investigative nature of computing will be modeled through hands-on activities that will allow students to gain insights about teaching and learning computing concepts in classroom settings. Students will examine a variety of computing tools, virtual environments, and other instructional resources that support learning across STEM disciplines. Students will learn that pedagogical methods in computer science require intellectual rigor in order to develop lessons that are relevant to secondary students and pertinent to our culturally diverse world today.

# AREAS OF EMPHASIS



1. Representing and connecting computer science concepts
2. Engagement with particular instructional strategies that foster inquiry-based teaching and learning
3. Eliciting, assessing, and understanding students' computer science content knowledge and computational thinking practices
4. Development of equity-based teaching practices in computer science education

# Required Reading

- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Bell, T., Alexander, J., Feeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Ben-Ari, M. (2004). Situated Learning in Computer Science Education. *Computer Science Education*, 14(2), 85-100. doi: 10.1080/08993400412331363823
- Eglash, R., Bennett, A., O'Donnell, C., Jennings, S., & Cintonino, M. (2006). Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom. *American Anthropologist*, 108(2), 347–362.
- Goldman, J., Shilton, K., Burke, J., Estrin, D., Hansen, M., Ramanathan, N., Reddy, S. Samanta, V., Srivastava, M., & West, R. (2009) “Participatory Sensing: A citizen-powered approach to illuminating the patterns that shape our world.” Woodrow Wilson Center for International Scholars.
- Goode, J., & Chapman, G. (2011). Exploring computer science (4<sup>th</sup> ed.). Los Angeles: Computer Science Equity Alliance.
- Goode, J., & Margolis, J. (2011). Exploring Computer Science: A Case Study of School Reform. *Trans. Comput. Educ.*, 11(2), 1-16.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: Urban youth learning programming with scratch*. Paper presented at the Proceedings of the 39th SIGCSE technical symposium on Computer science education, Portland, OR, USA.
- Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.
- National Research Council. (2000). *Inquiry and the National Science Education Standards: A Guide for Teaching and*

# Assignments

## **Reflective Journaling - Weekly**

- Each week as part of class, students will engage in reflective journaling in response to prompts posed by the instructors. These prompts will be related to the readings, class discussions, and instructional activities. Journals will be submitted to the instructor weekly for written feedback.

## **Computer Science Classroom Observation – Due Week 5**

- Students will visit a secondary computer science class and conduct an observation of teaching and learning. Students will write a 2-page single-spaced overview of this observation, including a description of the lesson's content, instructional delivery, how equitable classroom context will be created, student engagement, and intellectual rigor.

## **Planning and Delivering an Instructional Lesson – Due Weeks 7-9**

- In a small assigned group of students, develop and lead a one-hour instructional lesson on one of three recommended topics. This lesson should emphasize an inquiry-based approach to engage students in hands-on learning. Lessons should have a concluding component that links the computing concepts at hand to related mathematics, science, or computer science content. Topics include:
  - CS Unplugged
  - Scratch
  - Culturally situated design tools

WEEK	TOPIC	READINGS	ACTIVITIES
1	<b>Introduction to course</b>	<ul style="list-style-type: none"> <li>Syllabus</li> </ul>	<ol style="list-style-type: none"> <li>1) Discussion: What is CS, anyway?</li> <li>2) Activity: Room Picture Activity (ECS pp. 45-50) )</li> <li>3) Mobilize: Ethics and Mobilize assignment given</li> </ol>
2	<b>Equity and Access in CS Education</b>	<ul style="list-style-type: none"> <li>Stuck in the Shallow End: Jigsaw</li> </ul>	<ol style="list-style-type: none"> <li>1) Activity: Students talk in groups about assigned chapters and present to rest of class</li> <li>2) Activity: Stereotype threat (tbd)</li> <li>3) Mobilize: Forming good research questions and collecting appropriate data</li> </ol>
3	<b>Curricular Concepts in Computer Science</b>	<ul style="list-style-type: none"> <li>Barr &amp; Stephenson</li> <li>CSTA Model Curriculum</li> </ul>	<ol style="list-style-type: none"> <li>1) Activity: Jigsaw computing topics/levels from CSTA document to review and report out to class</li> <li>2) Assign groups for instructional lesson assignment and allow for group planning time</li> </ol>
4	<b>Inquiry in Computer Science</b>	<ul style="list-style-type: none"> <li>National Research Council</li> </ul>	<ol style="list-style-type: none"> <li>1) Activity: Model inquiry lesson and debrief as class (teacher/learner/observer model)</li> <li>2) Mobilize: Data Analysis</li> </ol>
5	<b>Moving to the Deep End: ECS</b>	<ul style="list-style-type: none"> <li>Goode &amp; Margolis</li> <li>Goode &amp; Chapman</li> </ul>	<ol style="list-style-type: none"> <li>1) Debrief CS classroom observations</li> <li>2) Discussion: Democratizing CS</li> <li>3) Mobilize: Data Analysis II</li> </ol>

WEEK	TOPIC	READINGS	ACTIVITIES
6	<b>Assessing Student Learning</b>	<ul style="list-style-type: none"> <li>Ben-Ari</li> </ul>	<ol style="list-style-type: none"> <li>1) Activity: Examining learning in ECS videos</li> <li>2) Activity: Designing a digital portfolio for a CS, mathematics, or science classroom</li> <li>3) Mobilize: Representing findings</li> </ol>
7	<b>Planning Inquiry Lessons</b>	<ul style="list-style-type: none"> <li>Goldman</li> </ul>	<ol style="list-style-type: none"> <li>1) Mobilize: Questions</li> <li>2) Group planning time for student-led lessons in weeks 7-10</li> </ol>
8	<b>Computing without Computers</b>	<ul style="list-style-type: none"> <li>Bell, et.al.</li> </ul>	<ol style="list-style-type: none"> <li>1) Group Instructional Lesson on CS Unplugged</li> <li>2) Class debrief of lesson</li> <li>3) Discussion: How would this lesson need to be adapted for secondary students?</li> </ol>
9	<b>Computing across Cultures</b>	<ul style="list-style-type: none"> <li>Eglash, et.al.</li> </ul>	<ol style="list-style-type: none"> <li>1) Group Instructional Lesson on Culturally Situated Design tools</li> <li>2) Class debrief of lesson</li> <li>3) Discussion: How would this lesson need to be adapted for secondary students?</li> </ol>
10	<b>Programming with Scratch</b>	<ul style="list-style-type: none"> <li>Maloney, et.al.</li> </ul>	<ol style="list-style-type: none"> <li>1) Group Instructional Lesson on Scratch</li> <li>2) Class debrief of lesson</li> <li>3) Discussion: How would this lesson need to be adapted for secondary students?</li> </ol>
11	<b>Mobilize &amp; Conclusion</b>		<ol style="list-style-type: none"> <li>1) Group Instructional Lesson on Mobilize</li> <li>2) Activity; Developing a plan for integrating computer science in STEM classes</li> <li>3) Debrief of course</li> </ol>



QUESTIONS?

# UTeach CS



Bradley Beth  
Calvin Lin  
Dept. of Computer Science  
February 3, 2012

# The UTeach Program

- Collaboration between College of Education and College of Natural Sciences
- Offers standard BS degree + teaching certification in 4 years
- Provides field work early and often
- Teaches inquiry-based learning

# A Success Story

- Has doubled the number of math and science teachers graduating from UT
- Produces teachers with higher 5-year retention rates
- Being replicated at 29 universities throughout the country

# A Sad Story



Only 7 CS certifiers in 14 years

→ UTeach begins (1997)  
→ First UTeach certifier graduates (2000)  
→ First UTeach CS certifier graduates (2001)

# The Problems

- Lack of awareness
  - Not aware of UTeach
  - Not aware that they get a true BSCS
  - Not aware that they could teach CS
  - Not aware of accurate salary/benefits for teachers
- Lack of support within UTeach
  - No UTeach CS degree option
  - No CS-specific pedagogy

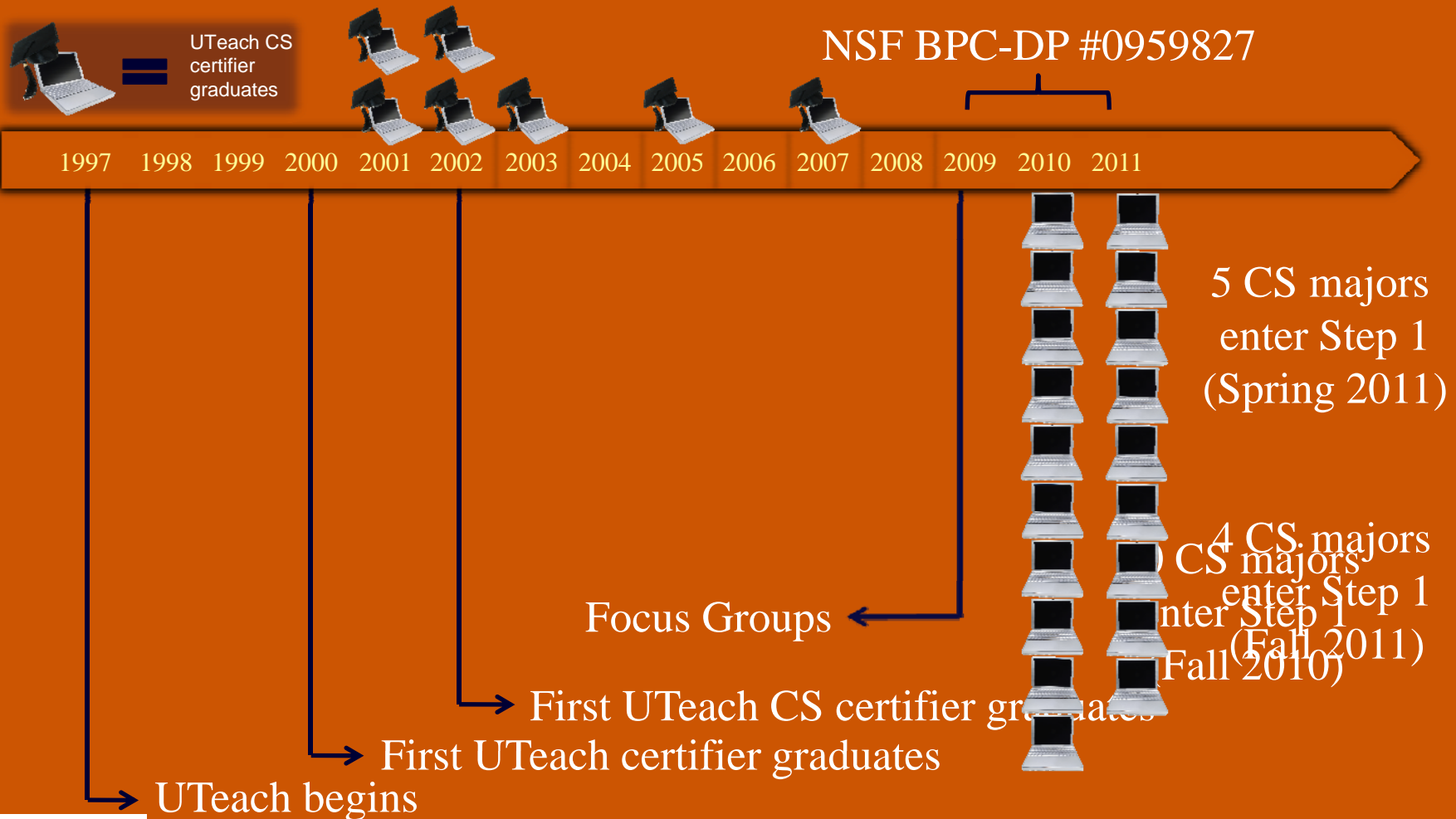
# Our Solutions: Recruitment

- Coordinate with advisors – both UTeach and CS
- Advertise in introductory CS courses
- Create personal connections

# Our Solutions: Retention

- New degree plans
  - BS in CS + CS certification
  - BS in CS + CS and Math certification
  - BS in Math + CS and Math certification (under review)
- Create CS-specific teaching supplements
- Cultivate a UTeach-CS student community

# The Rest of the Story



# New Solutions

- Can we explore alternate pathways?
  - Target other science majors (eg. Biology majors)
  - Target computing professionals

# Questions?