Jim Vallino, Rochester Institute of Technology

Program distinctions

- First undergraduate software engineering program in the US.
- Always been the largest with currently 400 undergraduate students; Also a small Masters program with around 25 students total.
- Only SE program in a standalone unit

My luxuries

- As an SE program, we have lots of time to cover SE material: 37 semester credits in 12 courses and one freshman seminar
- SE only faculty yields no pesky faculty who wonder about the whole SE thing

With the dream SE situation, why am I interested in what is in this first SE course?

- We provide a service course required for CS, CompE
- Taken in second year as preparation for co-op
- 350 – 400 students per year, 20 – 25 students per section
- For CS and CompE students, their experience is typical, i.e. all their SE is in this one (and only) course
- Our program is 17 years old and we have reworked this course 8 times several of them being significant rewrites

We have tried several flavors of this course.

- Straight waterfall: full use case requirements, functional specs, design documents, >10k lines of code – all in a 10 week quarter; First part of the course was document engineering → end of the course was a hackfest
- PSP/TSP-based → more document engineering
- Mostly an iterative approach now; struggle with tools
- A textbook has always been a problem
    - Selected a classic text thinking to use it throughout the SE curriculum.

    - Even though faculty agreed to this, none of them were willing to use the book in their downstream courses
    - Reality is that only half the students bought the book, and none of them read it.

We are looking at all our courses as part of conversion to a semester calendar. Have settled on several principles.

- Less is better
- Lightweight is better
- We should tailor the course for what is best for the CS and CompE students
- As long as we don't scare away the SE majors

Proposed topic because we thought we were the dummies who could not get this course right; lots of people have issues with it; CS2013 Ironman

I don't know the answer but I do have some ideas, and can describe some things that have not worked.

- Don't look at SE curriculum guide for detailed course content. There is too much stuff there. (Less is better)
- It's plain nuts to think that you can cover all of a classic text in this one and only course. Attempting to do that will turn into little more than a buzzword exercise
- A project is essential. Make sure that the class discussions tie closely to the projects; disconnects between lecture discussion and what goes on with the project will have the students asking "Why is this useful?"
- Tools are important to use but not to teach; concepts regarding why the tools are needed and the workflows that the tools will support are important to teach; use the students (TAs, student-created tutorials) to teach the details of tool use
- Waterfall is not the way to go unless your employer constituency is in narrow industrial segments, but even then consider this choice carefully; perhaps have an elective process course discussing methodologies
- Time (and points) for team to reflect on their practice is important

- Untrench the entrenched faculty.
    - Don't be straddled with being the only teamwork course
    - Don't be straddled with being the only communications course
    - Have "SE" topics covered in other courses (SDF, SE)
    - Have more than one SE course available