INTEGRATING ROLE-PLAY INTO SOFTWARE ENGINEERING COURSES

Tyson R. Henry
Janine LaFrance
Department of Computer Science
California State University, Chico
Chico, CA 95929-0410
530-898-5709
tyson@ecst.csuchico.edu, janine@ecst.csuchico.edu

Software Engineering

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software (IEEE Standard 610.12).

Teaching Software Engineering

Teach students how to apply a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software (IEEE Standard 610.12).

NDS: A Software Engineering Example

- NDS: Nuclear Detonation Detection System
- $100,000,000 contract to build ground system
- 60 programmers for 7 years
- Build it “right” this time (version 5, year 25)
- Formal software engineering process

Software Engineering?

- Shoe throwing
- I was waiting for Tyson to finish his part
- Runaway string library
- Maintaining thousands of pages of documents
- Iteration gridlock: just do it wrong for this year

Software Engineering (alternative definition)

The sociology and psychology, of getting a heterogeneous group of people (some of them less rational than others) to follow the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software.

Teaching Software Engineering

(1) Teach students the basics of software engineering and the current best practices.

(2) Teach students the personal and communication skills necessary to be productive in a software engineering organization.

Teaching SE Challenges

- Students don't believe it is important
  - Before SE we tell them that programming and theory are important (many are skeptical of the importance of theory)
- Software engineering activities are not as much fun as programming
- Sociological challenges can be enormous
- Students perceive software engineering as an pointless academic field
- Students find software engineering textbooks exceptionally boring

Preaching Software Engineering

- Software engineering is for large projects (not 5 students working for 16 weeks)
- It is very difficult to demonstrate that software engineering methods are anything but busy work (students believe they could produce more without them)

Preaching Software Engineering

- Often lectures end up as sermons on the worthiness of software engineering practice
- Student Reactions
  - This does not apply to me (just like the pumping lemma)
  - I'm a good programmer, so I'll get a job programming (instead of "software engineering")

The Solution?

- Reduce time lecturing
- Increase time students engage each other

Role-Play

- Have students perform role-play exercises in class
  - Structured (written instructions, written roles)
  - Short (1 – 2 class periods)
  - Designed to cause conflict
  - Groups answer a set of follow-up questions
  - Class discussion (debriefing)
  - Easy to download and use

Debriefing

- Most important part of exercise
- Class discussion lead by instructor
- Discuss answers to follow up questions
- Focuses on sociological and communication issues
- Gives instructor ability to direct students' introspection

Role-Play Advantages

- Fun (at least more fun than a lecture)
- Engaging (can't simply sit and daydream)
- Encourage introspection
- Memorable
- Enlighten students to the wide range of situations/problems

Role-Play Exercises

- Requirements Elicitation
- Risk Assessment
- Turnover
- Review
- Checkpoint

Requirements Elicitation

- On-line college advising system
- Roles
  - Software developers
  - Future customer (focus group members)

Risk Assessment

- Identify and create a mitigation plan for the top 10 risks in their semester project
- Roles
  - Group leader
  - Group member

Turnover

- Move the strongest member of each project group to a different group (just for the exercise)
- New groups much identify the cost of losing their member and develop a recovery plan

Review

- Each project group reviews a portion of their project (e.g. code segment, GUI component)
- Roles
  - Presenter (creator of artifact)
  - Reviewer

Checkpoint

- Stakeholder review of class project
- Roles
  - Upper management
  - Sales
  - Support personnel
  - Marketing
  - Developers

Student Reaction

- Good use of class time: 96%
- Should be used in future courses: 98%
- Relevant to course material: 99%
- Learned from the exercise: 89%
- Would rather have a lecture: 5%

52 students, 175 completed surveys

Student Reaction

- This was an eye-opener for all of my group
- It is important to plan for the unexpected
- I learned how important it is to communicate risks
- Puts working for a corporation into perfect perspective
- Communication is a big thing

Conclusion

- Students enjoy the role-play
- Has more impact than a lecture
- Shifts the focus from a sermon to an experience
- Gives students a taste of the sociological and communication problems they *will* face in their careers

Discussion

- Role-play is just one method for engaging software engineering students
  - Story telling
  - Games
  - Skits
  - Debates
  - Contests