Empirical Software Engineering in Industry Short Courses

David Janzen, Clark Turner
California Polytechnic State University
San Luis Obispo, CA

Hossein Saiedian
University of Kansas
Lawrence, KS

CAL POLY

Outline

- EBSE Background
- Pilot Experience
- Survey Results
- Challenges and Suggestions
- Late-Breaking News
- Conclusions

Evidence-Based Software Engineering (EBSE)

- Promises:
  - Demonstrate the efficacy of software practices, tools, and methods
  - Inform adoption decisions
- Tactics:
  - Controlled experiments
    - Laboratory (often with students)
    - Field (professionals in familiar/usual domain)
  - Case studies
  - Surveys (review existing studies and experiments)

EBSE Challenges

- Threats to validity with academic studies
  - Students less mature than professionals
  - Contrived application domains
  - Smaller project and (sometimes) team size
- Access to industry
  - Reluctance to participate in controlled experiments (IP/NDA, reputation, productivity)
  - Few want to try “bleeding edge” tools/practices
  - Ignorance (EBSE results, techniques, opportunities)

Opportunity

- Many professionals acquire new skills through Industry Short Courses
- Proposal:
  - Introduce EBSE techniques and results by conducting small experiments in Industry Short Courses

Expected Benefits

- Education on EBSE techniques and results
  - Goal is not to train researchers, but to inform practitioners of EBSE benefits and results
- Opportunities to conduct experiments
  - Lab experiments and surveys in training courses
  - Future field experiments in professional domain
- Teach analytical thinking
  - Consider alternatives (admits “no silver bullets”)
  - Compare EBSE results with personal experience

Pilot Experience

- Conducted quasi-controlled experiments in three industry short courses
  - On-site courses for two Fortune 500 companies

Experiment	Timeframe	Students	Assignment	Pairs/ Solo	Submissions TF	Submissions TL
TDD in Java	Fall 2005	15	Bowling	Both	3	3
C++ for C Programmers	Summer 2005	14	Bowling	Solo	2	4
TDD in Java	Fall 2006	14	Bowling	Pairs	2	3
TDD in Java	Fall 2006	14	ToDo List	Pairs	4	0

  - TDD in Java (2 day)
  - C++ for C Programmers (4 day with ½ day on day 4 devoted to TDD)

Context

- Series of leveled studies to compare test-first (TDD) and test-last approaches
- Hypothesis:
  - TDD improves internal software quality
    - Complexity, size, coupling, cohesion, testability
- Leveled:
  - CS1, CS2, Undergrad SE, Grad SE, Industry


TDD Instruction
Project/Phase 1
Project/Phase 2
Survey
CS1
CS2
SE (undergrad)
SE (grad)
Industry (in-training)
Industry (in-domain)
Test-First
Test-Last
No-Automated-Tests
This paper focuses on training experience

Ethical Considerations

- support the IEEE/ACM Software Engineering Code of Ethics section 7.02
  - "[a]ssist colleagues in professional development"
- Belmont Report states that "[a]pplications of the general principles to the conduct of research leads to consideration of the following requirements:
  - informed consent, risk/benefit assessment, and the selection of subjects of research."
  - Get approval from management in writing beforehand, get verbal or written approval from participants, allow them to not participate

Results of Study

- Caveat:
  - Results are of limited value due to common threats to validity
    - Small sample size (14 or 15 per study)
    - Short project duration (2-3 hours)
    - Immaturity with concepts (learning TDD)

Test Coverage

- Test-first programmers tended to achieve higher test coverage
- Differences were larger in larger projects

Experiment	Exercise	Approach	Coverage	
			Line	Branch
TDD Summer 2006	Bowling	TF	81%	79%
TDD Summer 2006	Bowling	TF	45%	0%
TDD Summer 2006	Bowling	TL	91%	87%
TDD Summer 2006	Bowling	TL	0%	0%
TDD Summer 2006	ToDo	TF	100%	100%
TDD Summer 2006	ToDo	TF	100%	100%
TDD Summer 2006	ToDo	TF	88%	75%
TDD Summer 2006	ToDo	TF	86%	76%
TDD Fall 2005	Bowling	TF	50%	19%
TDD Fall 2005	Bowling	TF	58%	55%
TDD Fall 2005	Bowling	TF	49%	30%
TDD Fall 2005	Bowling	TL	68%	63%
TDD Fall 2005	Bowling	TL	73%	80%
TDD Fall 2005	Bowling	TL	6%	0%
		Average TF	73%	59%
		Average TL	60%	58%

Survey Results 1

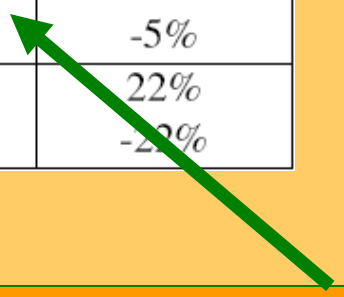
- Difference from Pre-experiment survey to Post-experiment survey
  - Importance of unit testing (Attitude)
  - Timing of writing unit tests (Timing)
    - Higher means they prefer earlier testing in process

Experiment	Direction	Attitude	Timing
TDD Fall 2005	%Increasing	62%	50%
	%Decreasing	0%	0%
C++ Summer 2005	%Increasing	0%	88%
	%Decreasing	13%	0%
TDD Summer 2006	%Increasing	9%	100%
	%Decreasing	9%	0%

Survey Results 2

- Choice of test-first or test-last programming (Choice)

Experiment	Choice	%Pre	%Post	%Difference
TDD Fall 2005	Test-First	67%	83%	17%
	Test-Last	33%	17%	-17%
C++ Summer 2005	Test-First	29%	33%	5%
	Test-Last	71%	67%	-5%
TDD Summer 2006	Test-First	60%	82%	22%
	Test-Last	40%	18%	-22%



Reluctance to adopt test-first by C programmers in C++ class

Challenges 1

- Difficult to calculate results quickly for immediate feedback
  - Used results from previous studies
  - Training did include test coverage tools so students reported their own test coverage results
- Short time frame
  - Difficult to create short yet meaningful exercises

Challenges 2

- Student reluctance to submit projects
  - Desire to master new concepts before showing others
- Integration with course concepts
  - Companies are paying you for training, not to conduct research

Suggestions

- Get approval from management in writing beforehand
- Get verbal approval from participants
- Allow trainees to not participate
- Identify trainee submissions with numbers
- Minimize “extra” time for conducting study
- Allow everyone to apply both(all) approaches/tools/methods

Late-Breaking News 1

- Conducted empirical study in middle-quarter of three quarter (nine-month) software engineering capstone
- Assigned students to perform study analysis in third quarter
- Results: students lacked enthusiasm about calculating and analyzing metrics, but gained appreciation for empirical approach

Late-Breaking News 2

- Met with representatives from both companies two weeks ago
- Both indicated desire to collaborate
- One indicated interest in participating in a new, larger empirical study

Conclusions

- Conducting empirical studies in industry short courses may:
  - Increase awareness of EBSE techniques and results
  - Increase opportunities for field experiments
  - Yield evidence on attitudes and learning curves
  - Yield supporting evidence for larger studies

Questions?