Are Students Representatives of Professionals in Software Engineering Experiments?

ICSE 2015

Iflaah Salman[1], Ayse Tosun Misirli[2], and Natalia Juristo[3]

[1] Department of Information Processing Science University of Oulu, Finland
[2] Faculty of Computer and Informatics Istanbul Technical University, Turkey
[3] Facultad de Informática Universidad Politécnica de Madrid, Spain

Presented by
Daniel Cruz

Agenda

- Introduction
- Research Method
- Study Context, Subjects and Analysis
- Results and Discussion
- Related Work
- Threats to validity
- Conclusion

1 Introduction

Introduction

- The external validity of experiments conducted with students is very often criticized
- There is an intense debate concerning the realism of the experiments, in the aspect of academia/industry subjects utilization

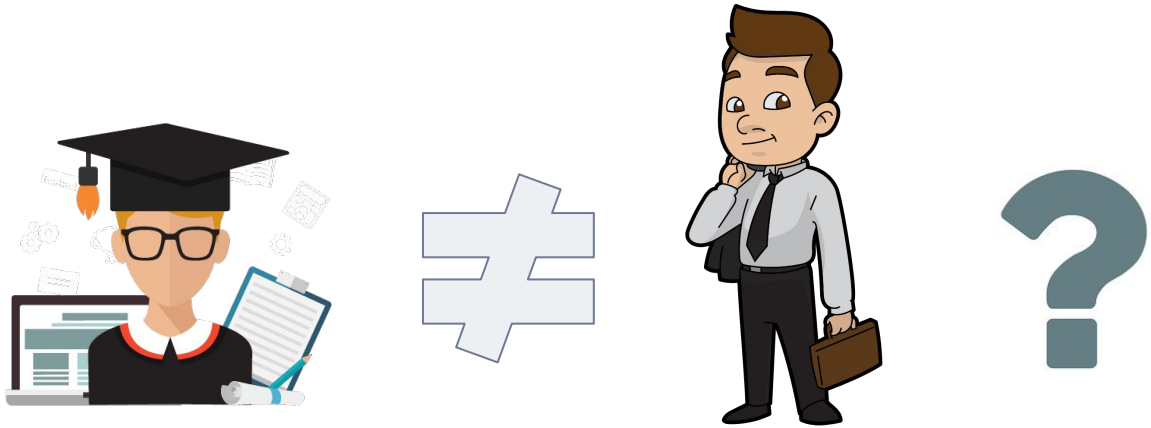
Introduction

- Professionals in a real software engineering (SE) environment is very expensive

- The use of students to conduct SE experiments is important to proceed with studies in which these professionals are not accessible

Introduction

- Controlled experiments with students are unrealistic in terms of environment, tasks and subjects (Sjoberg et al.)



Goal

"Compare students and professionals in order to gain an understanding of how representative students might be of professionals in SE experiments"

2 Research Method

Research Method

GQM

Analyze students as experimental subjects

For the purpose of comparison with professionals

With respect to the code quality of software tasks implemented by both subject groups

From the point of view of the students and professionals

In the context of a TDD experiment conducted in academia with students and in industry with professionals

Research Method

- Two experiments
  - Academic setting with students
  - Industrial setting with professionals
- Universidad Politécnica de Madrid (UPM), Spain
- Three offices of a multinational with over 25 years in the market, with 900 employees at 20 offices around the world

Research Questions

- RQ 1.1: How much does the **code quality** of a task[1] produced by **students** using **TDD** differ from the **code quality** of a task produced completed by **professionals** using **TDD**?

[1] code quality measured on a software system that is implemented for a particular TDD/TLD level

Research Questions

- RQ 1.2: How much does the **code quality** of a task[1] produced by **students** using **TLD** differ from the **code quality** of a task produced by **professionals** using **TLD**?

[1] code quality measured on a software system that is implemented for a particular TDD/TLD level

Variables of Empirical Study

- Independent variable
  - **Type of Subject**: student or professional

- Dependent variable
  - **Code Quality**: measured in terms of static code attributes (metrics) extracted from the source codes written by the subjects

TLD x TDD

- **TLD**: test-last development approach
  - Develop -> Write tests

- **TDD**: test driven development
  - Write tests -> Develop

3 Study Context,

Subjects and Analysis

Study Context

- This empirical study was conducted using the data from two experiments

- The research goal of both experiments was to observe the effects of TDD on quality and productivity

Study Context

- Experimental Design:

  one-factor two-level within-subjects

Study Context

- Experimental Design:

  one-factor two-level within-subjects

Development Approach

Study Context

- Experimental Design:

  one-factor two-level within-subjects

Development Approach

TLD e TDD

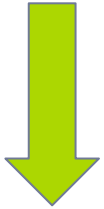
Study Context

- Experimental Design:

one-factor two-level within-subjects

Development Approach

TLD e TDD



Each subject realize all the tasks

Study Context

- Experimental Setup

	Industry Experiment	Academic Experiment
Experimental Design	ABB	ABBB
Objects/Tasks	MR, BSK, MP	MR, BSK, MP, S
Trainer	Same	
Time for the Tasks	Same	
Protocol	Five-day protocol	
Sampling	Convenience sampling	
Instrumentation	Eclipse IDE with JUnit testing framework; Java programming language; English as the training language	

Study Context

	Industry Experiment
Experimental Design	ABB
Objects/Tasks	MR, BSK, MP

3 Tasks
A - Using TLD
B - Using TDD on a toy example
B - Using TDD on a real-life application

	Academic Experiment
Experimental Design	ABBB
Objects/Tasks	MR, BSK, MP, S

4 Tasks
A - Using TLD - random assignment
B - Using TDD - random assignment
B - Using TDD - random assignment
B - Using TDD - random assignment

Study Context

	Industry Experiment
Experimental Design	ABB
Objects/Tasks	MR, BSK, MP

3 Tasks
A - Using TLD
B - Using TDD on a toy example
B - Using TDD on a real-life application

	Academic Experiment
Experimental Design	ABBB
Objects/Tasks	MR, BSK, MP, S

4 Tasks
A - Using TLD - random assignment
B - Using TDD - random assignment
B - Using TDD - random assignment
B - Using TDD - random assignment

* Tasks specifications link is broken:
https://www.dropbox.com/sh/i7i9im898db4u0n/AAA06ZXgJ1D

Study Context

	Industry Experiment
Experimental Design	ABB
Objects/Tasks	MR, BSK, MP

	Academic Experiment
Experimental Design	ABBB
Objects/Tasks	MR, BSK, MP, S

Real-life Application:

- MP - MusicPhone

Toy examples:

- BSK - Bowling Scorekeeper
- MR - Mars Rover
- S - Sudoku

Study Context - Protocol

Industry Experiment

- 1º Day: Unit Test Training and TLD Experiment Task (MR)
- 2º Day: TDD Training and TDD Experiment Task 1 (BSK)
- 3º Day: Study TDD at home
- 4º Day: Study TDD at home
- 5º Day: TDD Experiment Task 2 (MP)

Academic Experiment

- 1º Day: Unit Test Training
- 2º Day: TLD Experiment Task and TDD Training
- 3º Day: TDD Experiment Task 1
- 4º Day: TDD Experiment Task 2
- 5º Day: TDD Experiment Task 3

Study Context - Metrics

- 20 static code metrics
- Prest: open source metric extraction tool

Cyclomatic Density (CD)	Halstead Programming Time (HPT)
Decision Density (DD)	Maintenance Severity (MS)
Essential Density (ED)	Branch Count (BC)
Cyclomatic Complexity (CC)	Condition Count (Cnd.C)
Essential Complexity (EC)	Decision Count (DC)
Halstead Difficulty (HD)	Lines of Code (LOC)
Halstead Length (H.Len)	Total Operands (T.Oprnds)
Halstead Volume (HV)	Total Operators (T.Oprtr)
Halstead Level (H.Ll)	Unique Operands Count (U.Oprnd.C)
Halstead Programming Effort (HPE)	Unique Operators Count (U.Oprtr.C)

Subjects

- Academia
  - A course of an international graduate-level program
  - 17 Students
  - Many nationalities and languages

Subjects

- Industry
  - Large software organization
  - 24 Professionals
  - Many nationalities and languages, same in academic setting

Subjects

		Programming	Java Programming	Unit Testing	JUnit	TDD
Students	>10 years	0	0	0	0	0
	5-<=10 years	8	6	1	1	0
	2-<=5 years	5	7	2	2	0
	<2 years	4	4	14	14	17
	Total	17	17	17	17	17
Professionals	>10 years	10	3	2	1	0
	5-<=10 years	10	5	3	3	0
	2-<=5 years	4	9	14	8	5
	<2 years	0	7	5	12	19
	Total	24	24	24	24	24

Subjects

		Programming	Java Programming	Unit Testing	JUnit	TDD
Students	>10 years	0	0	0	0	0
	5-<=10 years	8	6	1	1	0
	2-<=5 years	5	7	2	2	0
	<2 years	4	4	14	14	17
	Total	17	17			
Professionals	>10 years	10	3			
	5-<=10 years	10	5		3	0
	2-<=5 years	4	9	14	8	5
	<2 years	0	7	5	12	19
	Total	24	24	24	24	24

More experienced in programming skills

Subjects

		Programming	Java Programming	Unit Testing	JUnit	TDD
Students	>10 years	0	0	0	0	0
	5-<=10 years	8	6	1	1	0
	2-<=5 years	5	7	2	2	0
	<2 years	4	4	14	14	17
	Total	17	17	17	17	17
Professionals	>10 years	10	3	2	1	0
	5-<=10 years	10	5	3	3	0
	2-<=5 years	4	9	14	8	5
	<2 years	0	7	5	12	19
	Total	24	24	24	24	24

More experienced in test skills

Subjects

		Programming	Java Programming	Unit Testing	JUnit	TDD
Students	>10 years	0	0	0	0	0
	5-<=10 years	8	6	1	1	0
	2-<=5 years	5	7	2	2	0
	<2 years	4	4	14	14	17
	Total	17	17	17	17	17
Professionals	>10 years	10	3	2	1	0
	5-<=10 years	10	5	3	3	0
	2-<=5 years	4	9	14	8	5
	<2 years	0	7	5	12	19
	Total	24	24	24	24	24

Subjects

		Programming	Java Programming	Unit Testing	JUnit	TDD
Students	>10 years	0	0	0	0	0
	5-<=10 years	8	6	1	1	0
	2-<=5 years	5	7	2	2	0
	<2 years	4	4	14	14	17
	Total	17	17	17	17	17
Professionals	>10 years	10	3	2	1	0
	5-<=10 years	10	5	3	3	0
	2-<=5 years	4	9	14	8	5
	<2 years	0	7	5	12	19
	Total	24	24	24	24	24

"We discussed with them and learned that although they received training on TDD, they never applied the technique in practice."

Analysis

- Normality tests on the metrics collected (Shapiro-Wilk)
- None of the metrics were normally distributed (p-value << 0.05)
- Non-parametric test, namely the two-sample Kolmogorov-Smirnov test (KS)

Analysis

- Check the general differences between two subject groups rather than differences in medians or variance

- Check if there are differences in terms of the minimum, maximum, median, mean and standard deviation of the metric data collected from the two subject groups

4 Results and Discussion

Data Reduction

- Removed data from subjects that did not attend to all experimental sessions
- 21 (out of 24) professionals
- 14 (out of 17) students

Descriptive Statistics

- For each experiment task and for each metric, computed the mean* for both groups, students and professionals

Metric	Students			Professionals		
	TLD	*TDD1*	*TDD2*	*TLD*	*TDD1*	*TDD2*
CD	1	1	1	1.5	1	1

* The complete descriptive statistics should be at https://www.dropbox.com/sh/7y7xv2h19p4qnas/AAD2eXuhbd, but it is broken

Descriptive Statistics

Metric	Students			Professionals		
	TLD	*TDD1*	*TDD2*	*TLD*	*TDD1*	*TDD2*
LOC	4	5	4	7	4	5
T.Oprnds	1	2	2	6	1	2
T.Oprtr	2	2	2	7	2	2
U.Oprnd.C	1	2	1	5	1	1
U.OprtrC	1	2	2	4	1	2
HD	0.5	1	1	2	0.5	1
H.Len	3	4	4	13.5	3	4
CC	4	4	4	3	3	4

Descriptive Statistics

- Code produced by professionals implementing in TLD fashion is easier to maintain
- Code produced by professionals in TDD fashion is harder to maintain
- Students produced smaller methods (LOC) than professionals in TLD task
- In the TDD1 task, students produced bigger methods (LOC) than professionals

Hypothesis Testing

- Kolmogorov-Smirnov (KS) tests on the minimum, maximum, mean, median and standard deviation values of each metric
- Total of sample comparisons: 5 (values) x 3 (Tasks) x 20 (metrics)

$$H_0CD: P\_TDD = S\_TDD$$

Null Hypothesis: the cyclomatic density of a TDD task implemented by professionals and a TDD task implemented by students is the same

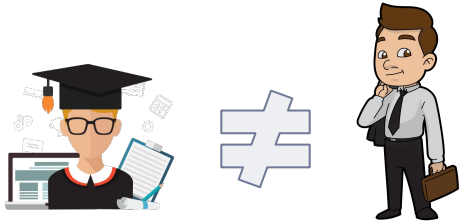
Hypothesis Testing

- Two approaches to compare:
  1. Code quality for all the tasks implemented by both groups based on the **experiment treatments** (TLD, TDD 1, TDD 2)

  2. Code quality for the tasks implemented by both groups based on **treatment/task combinations**, for example, TLD on MR system

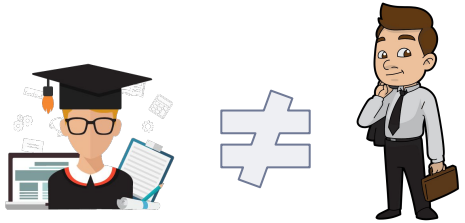
Hypothesis Testing

1. Comparison in Terms of Experiment Treatment
   - For **TLD**, 15 (out of 20) metric hypotheses were **rejected**, i.e, **75%** of the code quality metrics for the two subject groups follow different distributions
   - 55 out of 100 considering all the five metric values



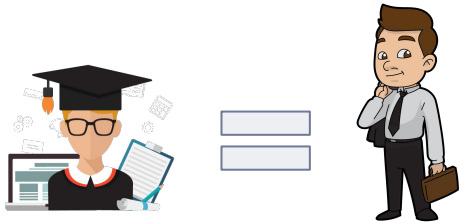
Hypothesis Testing

1. Comparison in Terms of Experiment Treatment
   - For **TDD 1**, 12 (out of 20) metric hypotheses were **rejected**, i.e, **60%** of the code quality metrics for the two subject groups follow different distributions
   - 49 out of 100 considering all the five metric values



Hypothesis Testing

1. Comparison in Terms of Experiment Treatment
   - For **TDD 2**, 7 (out of 20) metric hypotheses were **rejected**, i.e, **35%** of the code quality metrics for the two subject groups follow different distributions
   - 53 out of 100 considering all the five metric values



Hypothesis Testing

2. Comparison in Terms of Treatment-Task Combination
   - 3 Tasks linked to 3 Treatments:
     - MR - TLD
     - BSK - TDD 1
     - MP - TDD 2

Hypothesis Testing

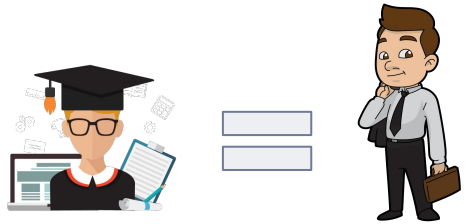
2. Comparison in Terms of Treatment-Task Combination
   - **The results differ substantially from comparisons based on treatment only**

Hypothesis Testing

2. Comparison in Terms of Treatment-Task Combination
   - For **MR-TLD**, 1 (out of 20) of metric hypotheses could **be rejected**, i.e, **5%** of the code quality metrics for the two subject groups follow different distributions
   - 10 out of 100 considering all the five metric values

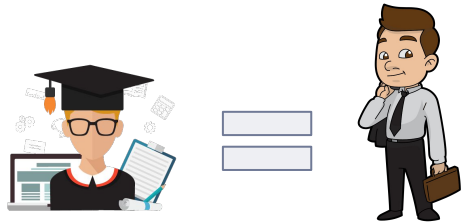
Hypothesis Testing

2. Comparison in Terms of Treatment-Task Combination
   - For **BSK - TDD 1**, 0 (out of 20) of metric hypotheses could **be rejected**, i.e, **0%** of the code quality metrics for the two subject groups follow different distributions
   - 1 out of 100 considering all the five metric values



Hypothesis Testing

2. Comparison in Terms of Treatment-Task Combination
   - For **MP - TDD 2**, 3 (out of 20) of metric hypotheses could **be rejected**, i.e, **15%** of the code quality metrics for the two subject groups follow different distributions
   - 40 out of 100 considering all the five metric values



Discussion

- In TLD case, a reason for the difference between the groups might be the experience of subjects, as professionals produce higher code quality in terms of modularity and complexity
- However, the effect observed in this context could also be due to tasks, since professionals worked on only one task (MR), while students worked on four, including a difficult one (MP)

Discussion

- For TDD 1 case, both subject groups appear to act similarly during their first implementation

- For TDD 2 case, they also performed similar, except for the fact that students produced more LOC and used more operators and operands

Discussion

- Most of the tests in the treatment-task comparison fail to reveal any differences between the two subject groups

- Differences among tasks seem to have a bigger impact on code internal quality than the subject type

Discussion

- The sample sizes for student data in this second comparison were quite small (5~7)

- Cohen's d power analysis for statistical tests confirms the findings

Discussion

- “How much does the code quality of a task implemented by a professional differ from that of a task implemented by a student? "
- For TLD and TDD applied on certain tasks, there are differences between the code quality of a task implemented by a student and a professional in terms of some metrics, like CC and LOC

5 Related Work

Related Work

Comparing Students and Professionals

- Several studies report the rates of participation of students and professionals in SE experiments
- Höfer et al., report that 60% of experimental studies in SE used students, whereas 22% employed professionals, and only 14% used both subject groups
- Very few studies have been specifically carried out to compare the performance of students and professionals

Related Work

Study	Objective/Aim	Context	Type	Subjects	Results
Porter & Votta [6]	-Comparison of the performance of students and professionals -Extension of the external credibility of a previous study	Fault detection techniques for software requirements specification inspections	Experiment with professionals	18 professionals48 students	All hypothesis tests revealed the difference in the performances of students and professionals.
Host et al. [5]	Difference in the performance of students and professionals	Factors affecting the project lead time	Experiment	18 professionals 26 students	-Minor differences in the conception of students and professionals for the factors - No significant differences between students and professionals when compared with actual effect of factors

Sample of 2 from 6 studies found

Related Work

Comparing Students and Professionals

- This study investigates the difference between students and professionals in the context of a technology-oriented experiment that observes the effects of the TDD approach on code quality

- And don't collect personal opinions from the two subject groups, but measure the artifacts that they produce and compare the code quality achieved by each group

Related Work

Experimentation on Test-Driven Development

- Identified previous studies investigating the effects of TDD on internal code quality rather than external code quality

- Of the recent systematic reviews on TDD literature, Munir et al. show that 15 out of 48 experimental studies (around 31%) investigated the effect of TDD on internal code quality

Related Work

Experimentation on Test-Driven Development

- Munir et Al. found 10 metrics used for measuring internal code quality In TDD experiments

Metrics for Internal Code Quality
Method / Statement/ Condition Coverage
Branch Coverage
Nested Block Depth
Cyclomatic Complexity
Number of Parameters
Coupling between Objects
Information Flow
Weighted Class per Method
Lack of Cohesion Metrics
Mutation Score Indicator

- In this study, 20 metrics are used

6 Threats to validity

Threats to validity

Internal Validity

- Subjects Selection: subjects might not be representative of the general community
- Convenience Sampling: was not possible to form a randomly generated sample for many reasons
- Diffusion or imitation of treatments: because random assignment, a student could transfer knowledge to another one between days of protocol

Threats to validity

External Validity

- The scope of the study findings would be limited to study conditions: TLD and TDD as technology and subjects' skills (Java programming and Unit Tests/JUnit)

Threats to validity

Construct Validity

- Mono-operation bias: used only a single version of a each system at a single point in time, which was addressed by assigning multiple objects to multiple subjects
- Mono-method bias : addressed by quantifying code quality through 20 static code attributes that are well known in quality prediction research

Threats to validity

Conclusion Validity

- Violated assumptions of statistical tests: avoided by applying normality test and the correct non-parametric test for data analysis
- Repeated hypothesis testing: potential risk of accumulated type I error (FP)

7 Conclusion

Conclusion

- The study investigated if students are representative of professionals as experimental subjects in the context of a test-driven development experiment
- When subjects apply an incremental test-last approach (TLD), a well known approach for them, there is a difference
- When they apply a new technology for the first time, TDD in this case, both subject groups perform similarly

Conclusion

- Support previous findings that neither of the subject groups performs better than the other when they apply a new technology during experimentation
- This kind of study incrementally build knowledge and contributes with empirical evidences
- A key condition for subjects' characterization might be the experience rather than which type of site (Academic or Industry) they belong to


Thanks!
Any questions?