A Topics Course in Empirical Software Engineering: Bridging Research and Practice

Week 3, Sept 25th 2020

Instructor: Margaret-Anne (Peggy) Storey

Guest: Dr. Rashina Hoda

Next week! Literature reviews and Knowledge in SE

Invited guest: Dr. Elise Paradise (U of T/UVic)

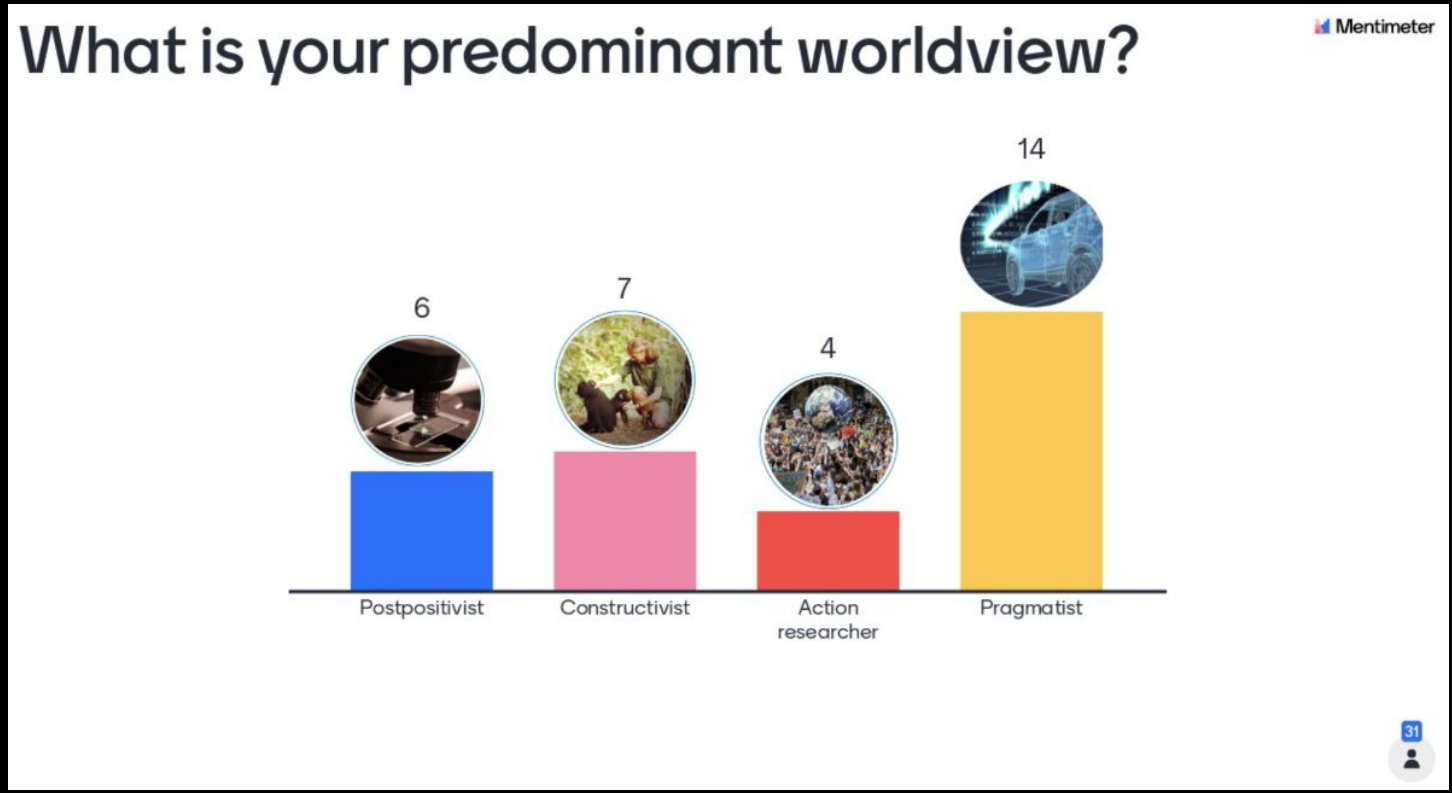
Preparing for class: See the readings/podcasts in GitHub

Recap from last week: Introduction to Empirical Software Engineering

Greg Wilson interview

Video of his podcast is ready to post (confirming it is ok with him and *You* it is ok to share (please message me privately if you have *any* concerns) as we did turn on our videos at the very end.

From last week: Your worldview...



Last week: Beliefs and Evidence “replication”

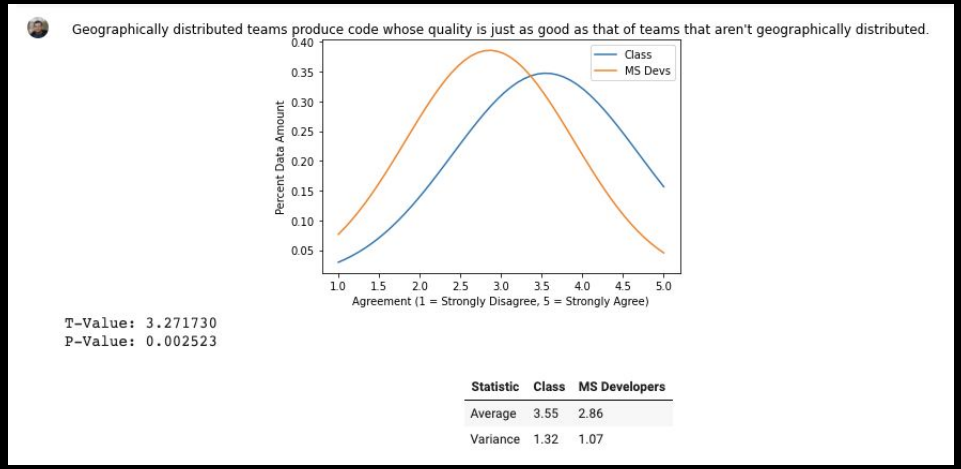
Sheet1

Question
Fixing defects is riskier (more likely to cause future defects) than adding new features.
Code quality (defect occurrence) depends on which programming language is used.
Geographically distributed teams produce code whose quality is just as good as that of teams that aren't geographically d
When it comes to producing code with fewer defects, specific experience in the project matters more than overall progran
Stronger code ownership (fewer people owning a module or a file) leads to better code quality.
Merge commits are buggier than other commits.
Components with more unit tests have fewer customer-found defects.
More defects are found in more complex code.
Using assertions improves code quality.
Using static analysis improves code quality.
Coding standards help improve code quality.
Code review improves code quality.

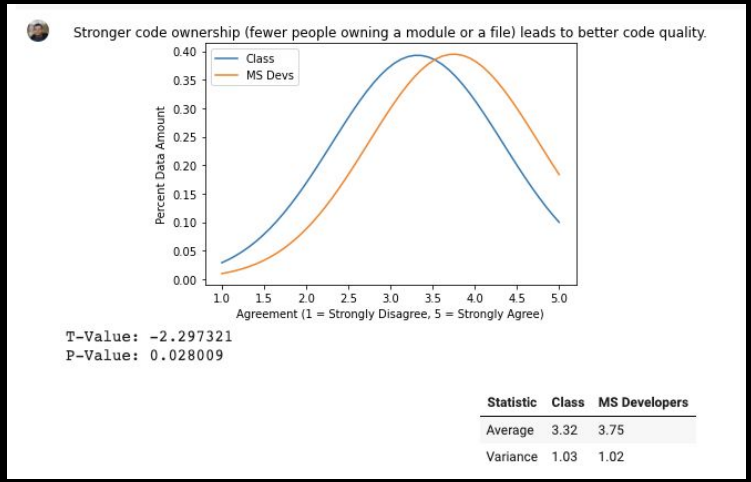
Question	Score	Variance
Code quality (defect occurrence) depends on which programming language is used [46]	3.17	1.16
Fixing defects is riskier (more likely to cause future defects) than adding new features [34, 48]	2.63	1.08
Geographically distributed teams produce code whose quality (defect occurrence) is just as good as teams that are not geographically distributed [29, 6]	2.86	1.07
When it comes to producing code with fewer defects specific experience in the project matters more than overall general experience in programming [39]	3.5	1.06
Well commented code has fewer defects [52]	3.4	1.05
Code written in a language with static typing (e.g., C#) tends to have fewer bugs than code written in a language with dynamic typing (e.g., Python) [46, 15]	3.75	1.02
Stronger code ownership (i.e, fewer people owning a module or file) leads to better software quality [7, 57, 15]	3.75	1.02
Merge commits are buggier than other commits.	3.4	0.97
Components with more unit tests have fewer customer-found defects [22].	3.85	0.95
More experienced programmers produce code with fewer defects. [34, 39]	3.86	0.94
More defects are found in more complex code. [25]	4.0	0.93
Factors affecting code quality (defect occurrence) vary from project to project. [59, 42]	3.8	0.92
Using asserts improves code quality (reduces defect occurrence) [4, 3]	3.78	0.89
The use of static analysis tools improves end user quality (fewer defects are found by users) [53, 58]	3.77	0.87
Coding standards help improve software quality [8]	4.18	0.79
Code reviews improve software quality (reduces defect occurrence) [38]	4.48	0.64

Recap of last lecture…. experiment replication

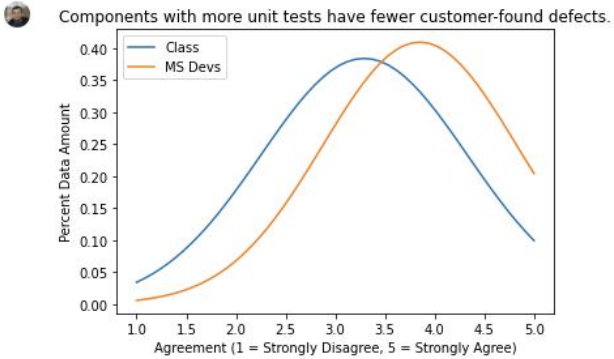
https://colab.research.google.com/drive/1crG2LZ6hpUaJvIrVP4WK60ouZ0q0-DLS?usp=sharing#scrollTo=ACG1a4p_jpfi



Statistic	Class	MS Developers
Average	3.55	2.86
Variance	1.32	1.07



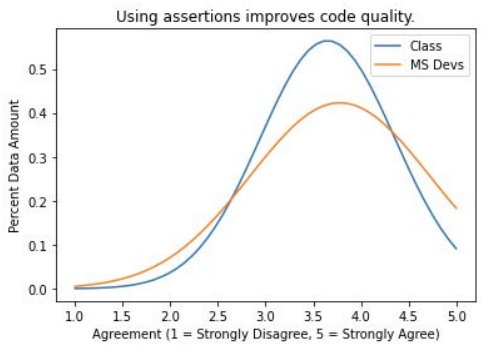
Statistic	Class	MS Developers
Average	3.32	3.75
Variance	1.03	1.02




```
T-Value: -2.930247
P-Value: 0.006108
```

Statistic	Class	MS Developers
Average	3.29	3.85
Variance	1.08	0.95

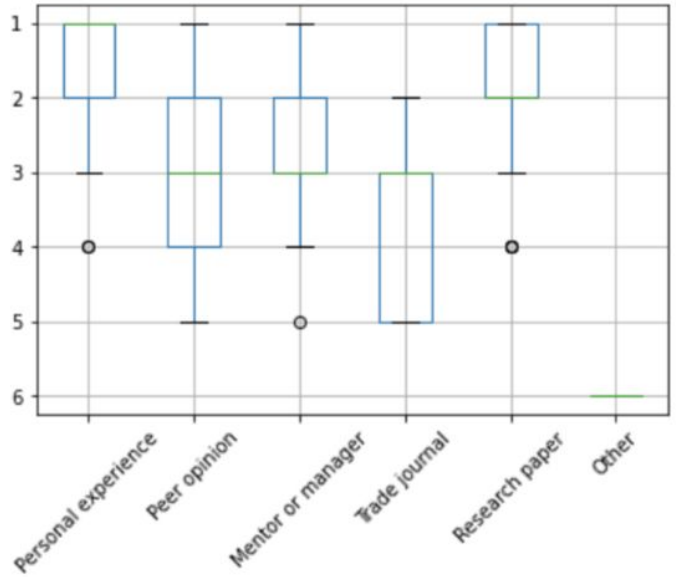
```
compare_normal_dist(statement_data_raw.loc[8, : ])
conduct_ttest(statement_data_raw.loc[8, : ])
```

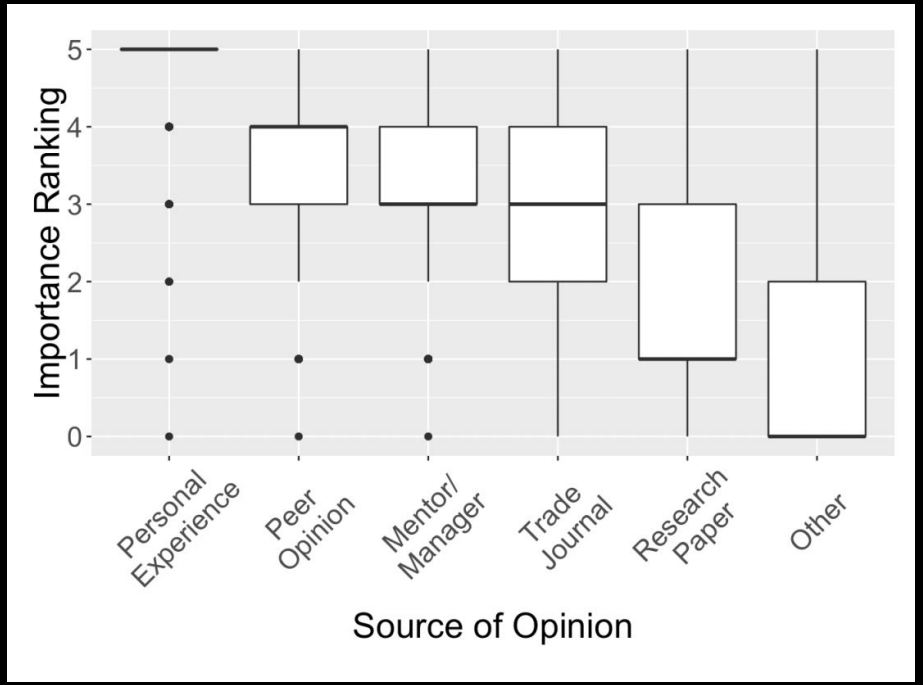



```
T-Value: -0.976945
P-Value: 0.335091
```

Statistic	Class	MS Developers
Average	3.65	3.78
Variance	0.5	0.89

```
sources_data_raw.boxplot(rot=45)
plt.gca().invert_yaxis()
```



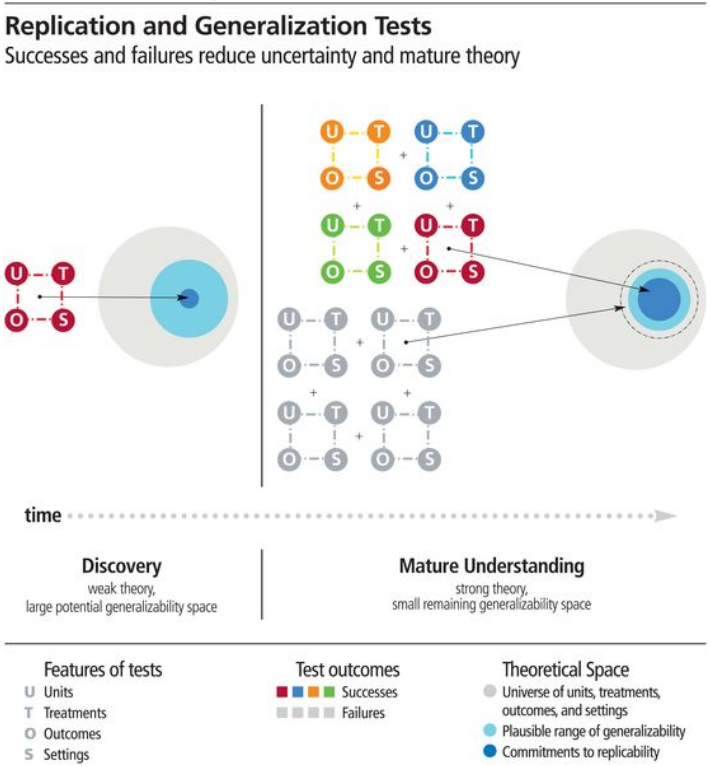


Food for thought: was this really a “replication”

“According to common understanding, replication is repeating a study’s procedure and observing whether the prior finding recurs. This definition of replication is intuitive, easy to apply, and ***incorrect****.”*

https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000691

Fig 1. There is a universe of distinct units, treatments, outcomes, and settings and only a subset of those qualify as replications—a study for which any outcome would be considered diagnostic evidence about a prior claim.



Nosek BA, Errington TM (2020) What is replication?. PLOS Biology 18(3): e3000691. https://doi.org/10.1371/journal.pbio.3000691
https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000691

PLOS BIOLOGY

Plan for rest of today (Sept 25th)

Part 1: Design science discussion (carry over...)

Part 2: Research methods and strategies

(Activity/Break)

Part 3: Dr. Rashina Hoda, Grounded theory

A Topics Course in Empirical Software Engineering: Bridging Research and Practice

Week 3, Part 3, Sept 25th 2020

Instructor: Margaret-Anne (Peggy) Storey
Guest: Rashina Hoda

Rashina Hoda