



Empirical SE (version 2.0)

tim@menzies.us
CS, WVU

<http://menzies.us>
July28, 2011

Small vs big science



High school science experiment

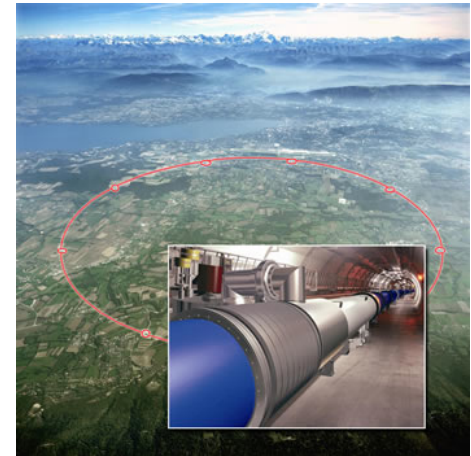
- Small teams
- Minimal infrastructure
- Minimal automated support, so even small scale data collection is complex and time consuming
- Able to probe a small number of well-defined, pre-existing hypotheses



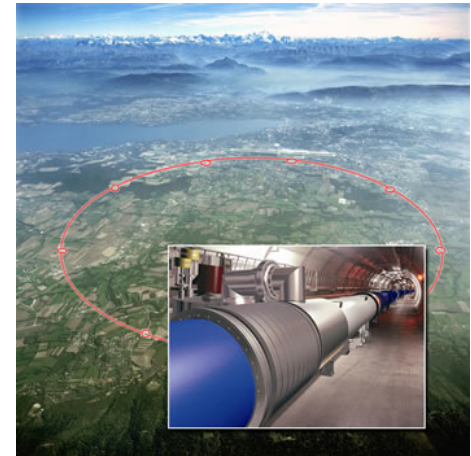
Empirical SE : as is

Large Hadron Collider

- International consortium
- Elaborate infrastructure
- Massive automatic support for large scale data collection
- Potential to probe deeper into core issues than ever before



Empirical SE : to be



- What am I doing to mature empirical SE?



The sorry state of empirical SE

- [Robles10]: review of 7 years of the MSR conference
 - Only 2 of 154 papers provided data & tools needed for replication
- [Zannier06]: review on “empirical” ICSE papers
 - Only 2% compared methods from multiple researchers.
- [Neto07]: review on model-based testing
 - 85 papers described 71 distinct approaches.
 - Few with actual experiments
- [Menzies08]: review of 100 suggested methods for V&V
 - No publications assessing relative merits of pairs of methods.

-
- 1 Zannier, C., G. Melnik, and F. Maurer. 2006. “On the Success of Empirical Studies” in the International Conference on Software Engineering. *Proceedings of the 28th international conference on software engineering*.
 - 2 Neto, A., R. Subramanyan, M. Vieira, G.H. Travassos, and F. Shull. 2008. “Improving Evidence About Software Technologies: A Look at Model-Based Testing” .*IEEE Software* 25(3): 10–13
 - 3 Menzies, T., M. Benson, K. Costello, C. Moats, M. Northey, and J. Richardson. 2008. “Learning Better IV & V Practices”. *Innovations in Systems and Software Engineering* 4(2): 169–183.
 - 4 G. Robles. Replicating MSR: A Study of the Potential Replicability of Papers Published in the Mining Software Repositories Proceedings. *Proceedings of the Working Conference on Mining Software Repositories, 2010*.

The PROMISE conference: Repeatable, ?improvable, ?refutable SE experiments



- Founded in 2005 by Tim Menzies & Jelber Sayyad
 - Goal: more SE results, faster
- Data mining on SE data
 - Authors asked to submit data from their paper.
- An on-line repo:
 - <http://promisedata.org/data>
 - no passwords,
 - no robots.txt

“PROMISE? It’ll never work”

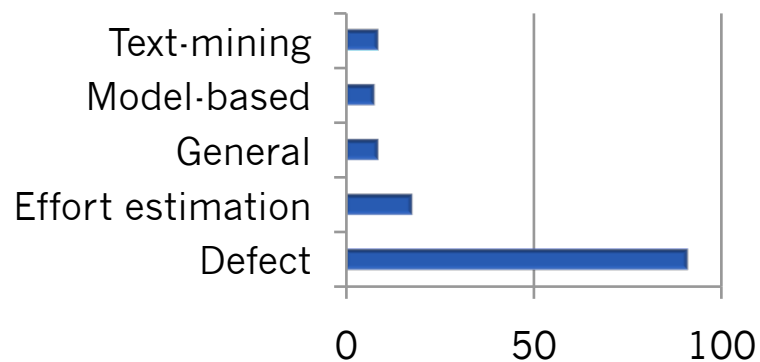


- Lionel Briand, Simula, 2005
 - “No one will give you data”



- Wrong!
 - To collect data,
 - Just ask for it

- 2011 data sets in PROMISE



- Why organizations give us data:
 - Civic duty (NASA)
 - Open source projects
 - Conferences (ESEM, MSR)
 - Misc
- Why researchers give us data:
 - Civic duty
 - Archiving
 - Peer recognition;
 - e.g. “tim menzies” at academic .research.microsoft.com
 - Last 5 years
 - Ranked #46 out of 56,000+ researchers
 - Papers ranked #37, #149 in 44,000+ papers

Report card

- What's been learned so far from PROMISE?



1

Tune your learners to the business goals

2

Seek local, not global, lessons learned

3

Your (filtered) data can be applied to my projects

4

Adjust your goals to the data

1 Tune your learners to the business goals



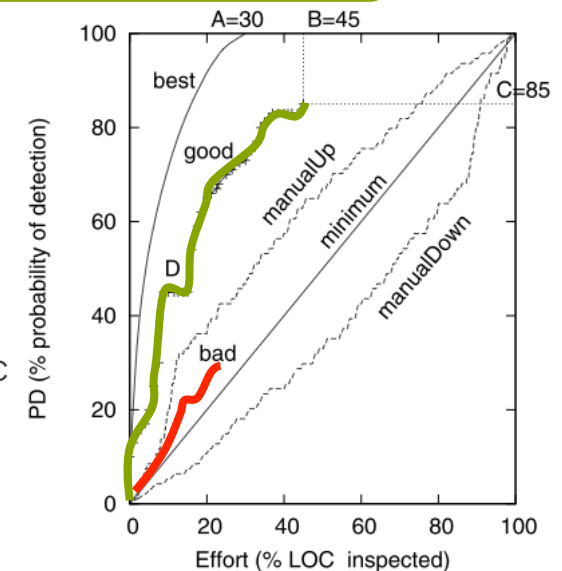
Evaluation methods : misdirected?

- Standard data miners evolved to maximize accuracy
 - More recently: cost-sensitive learning¹ or cost curves²
- Generate models that are orthogonal to business needs
- E.g. “inspection optimization”
 - A defect detector triggers
 - A human now reviews code
 - How to offer them the least code to find the most bugs?
 - Recall / LOC

Menzies et al 2010³

WHICH: Rules scored and sorted

- Combine 2 rules near top of sort
- Score, add back to sort.
- Note: scoring can be domain specific e.g. recall / LOC.
- Take code that triggers **WHICH's** rules, sort up by LOC
- Find bugs much earlier than C4.5 (**e.g. bad**)
- Moral: tune learners to business goals



- Yue Jiang and Bojan Cukic. 2009. Misclassification cost-sensitive fault prediction models. In *Proceedings of the 5th International Conference on Predictor Models in Software Engineering (PROMISE '09)*.
- Yue Jiang, Bojan Cukic, Tim Menzies, "Cost Curve Evaluation of Fault Prediction Models," *Software Reliability Engineering, International Symposium on*, pp. 197-206, 2008 19th International Symposium on Software Reliability Engineering, 2008
- Tim Menzies, Zach Milton, Burak Turhan, Bojan Cukic, Yue Jiang, Ayse Basar Bener: Defect prediction from static code features: current results, limitations, new approaches. *Autom. Softw. Eng.* 17(4): 375-407 (2010)

2 Seek local, not global, lessons learned



Conclusion instability

- Vic Basili¹
 - exploring empirical SE for over thirty years.
- Says that empirical SE is healthier now than the 80s
- Acknowledges that
 - results thus far are incomplete
 - few examples of methods that are demonstrably useful on multiple projects.

Menzies, Zimmermann, et al 2011²

1. Learn *global model* from all data
2. Cluster SE defect or effort data.
 - For each cluster C1 ask “which neighboring cluster C2 do you most envy” (lowest defects or effort)
 - Learn *local model* on C2
 - Test *local* and *global* on C1

Results:

- local models are *different* and *perform better* than global models.

Conclusion:

- Locality is a feature of SE data
- Don't generalize beyond local clusters
- Monitor a project, alert when leaves cluster
- Stop using cross-val to assess SE models

¹ V. Basili, personnel communication, 2009

² T. Menzies, A. Butcher, A. Marcus, T. Zimmermann, D. Cok “Local vs Global Models for Effort Estimation and Defect Prediction”, IEEE ASE 2011

3 Your filtered data can be applied to my projects



Old answer

- Brendan Murphy¹ :
 - Generality requires standards, inhibits fast-paced innovation
- Tom Zimmermann²:
 - We tried it, it didn't work: 600+ pairs of (project1,project2)
 - In only 4% of the pairs did defect predictors learned from project1 work on project2
- Barbara Kitchenham³:
 - evidence inconclusive

New answer (from PROMISE repo data)

- Yes we can.
 - The trick is to filter the training data.

1 T. Zimmermann, N. Nagappan, et al, "Cross-project defect prediction," *ESEC/FSE'09, August 2009*.
2 B. Murphy, "Why can't predictive models be generalized?" *ESE journal, 2012, to appear*
3 B. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus within company cost estimation studies: A systematic review," *IEEE Trans.Softw. Eng.*, vol. 33, no. 5, pp. 316–329, 2007,

3 Your filtered data can be applied to my projects (more)



11

[TurhanMenzies09]¹

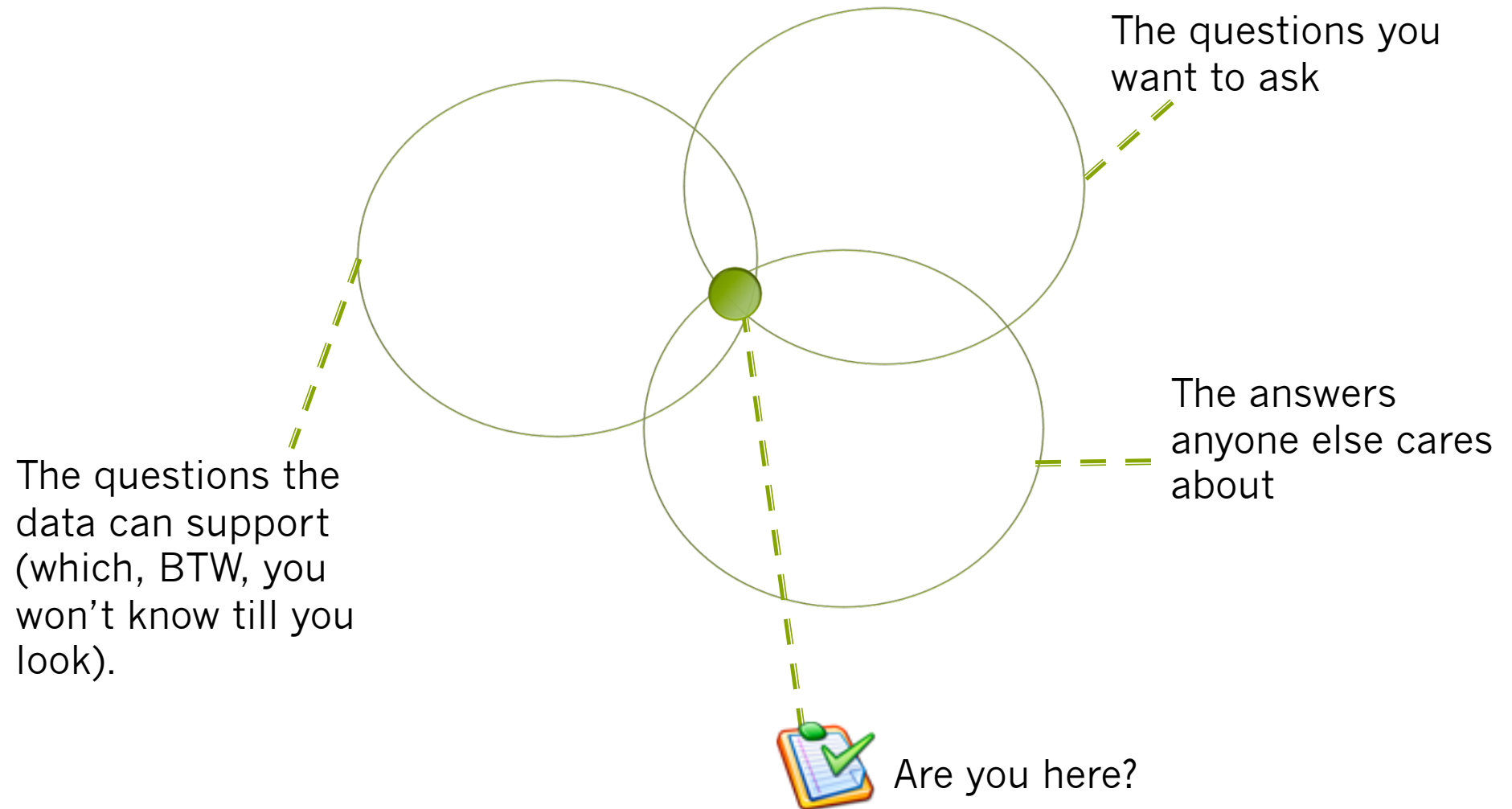
- Given data sets from different projects and company “A,B,C,etc”
 - Predict effort or defects for “A”
 - Train, after selecting from “B,C,etc”
 - *relevant* examples
 - *non-perplexing* examples
- *Finding relevant examples*
 - For members of “A”,
 - build a training set from the 10 closest neighbors in “B,C,etc”
 - Performs (nearly) as well as training on local data

[KocaguneliMenzies11]²

- *Avoiding perplexing examples*
 - Hierarchical clustering of all training data “A,B,C,etc”.
 - Prune high variance sub-trees
 - Estimate using the remainder
 - After variance pruning, there is no “cross” data.
 - “A” (within) selected with equal probability to “B,C,etc” (cross)

1 B. Turhan and T. Menzies and A. Bener and J. Distefano. “On the Relative Value of Cross-Company and Within-Company Data for Defect Prediction” *Empirical Software Engineering* pages 278-290 2009 .
2 Ekrem Kocaguneli, Tim Menzies, “How to Find Relevant Data for Effort Estimation?”, *ESEM 2011*

4 Adjust your goals to the data



4

Adjust your goals to the data



13

Parable v1.0

- One night, I meet a drunk searching the street.
 - "Can't find my keys", he said. .
 - "Are you sure you lost your keys here?" I asked.
 - "No" the drunk replies "I lost them over in that alley but there's no light there."
- Moral (v1.0):
 - Don't chase data, merely because it is easiest to collect

Parable v2.0

- As before, then...
 - "A ha!" shouted the drunk.
 - "Found the keys?" I asked
 - "Better! Tire tracks to bus stop!"
 - So he did not drive home drunk



- Moral (v2.0):
 - Study your data,
 - Then revise your goals



14

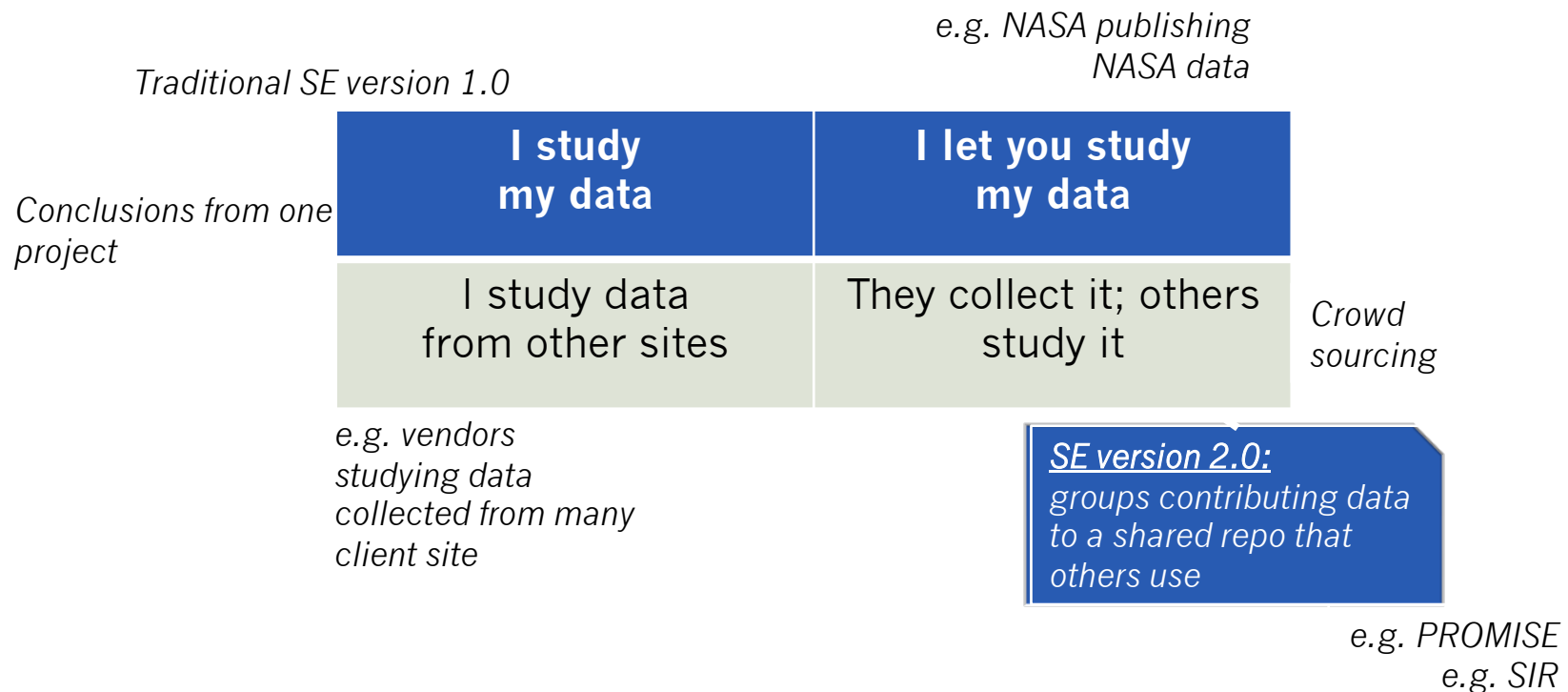
Questions?
Comments?



Empirical SE (Version 2.0)

Induction studies (e.g. data miners applied to PROMISE data)

- Old data
- New data miners
- New insights



What is wrong with Empirical SE (version 1.0)?



16

1. Systematic literature reviews


- Read & synthesize what other people have done ¹
- To date, inconclusive ²

2. Case studies

- Watch, don't touch
- Slow, labor intensive
- Reproducibility?
- Generality beyond the studied project?

3. Experiments

- Controlled manipulation of a few aspects of a project
- Usually, simple projects
- Industrial experimentation very expensive ³



Too slow to
keep up with
changes in SE

1 B. Kitchenham, O Pearlbrereton, D. Budgen, et al "Systematic literature reviews in software engineering – A systematic literature review" *Information and Software Technology*, 51(1), pages: 7-15, 2009

2 Budgen & Kitchenham, "Is evidence based software engineering mature enough for practice & policy?" *SEW-33, Skvde, Sweden, 2009*

3 Bente C.D. Anda, Dag I.K. Sjøberg, and Audris Mockus. "Variability and reproducibility in software engineering: A study of four companies that developed the same system." *IEEE TSE*, 35(3), May/June 2009.