Dossier
Tim Menzies
2015-2016

Department of Computer Science
North Carolina State University
email: tjmenzie@csc.ncsu.edu;
URL: http://menzies.us

STATEMENT OF MUTAL EXPECTATION

Tim Menzies, Computer Science, NC State

I. Introduction

A. Statement of Mutual Expectations (SME)

i) General Departmental Expectation – Responsibilities and contributions of CSC faculty conform to relevant departmental, college, and university rules, policies and regulations, including departmental Teaching Load, Release Time, and Definitions policies (http://www.csc.ncsu.edu/department/policies/index.php) ; departmental, college and university Reappointment Promotion and Tenure (RPT) and Post-Tenure Review (PTR) policies, etc. Faculty work in collegial manner. Load follows departmental load policy and distribution. Excellence is expected. Evidence of good teaching, research, service and other activities is required. Faculty are evaluated, based on the departmental and college RPT and PTR rules using individual consideration but with due attention to total and effective service to the institution. Faculty pro-actively mentor junior colleagues, and participate in faculty recruiting and other activities of import to the department, including those that increase and propagate good national and international standing, ranking and reputation of the department. Department will support faculty according to their contributions, departmental mission and scope, policies, regulations and rules, and within the scope of funding and other resources available to the department. This document may contain short-term and long term goals, actions and information. It will be updated as needed[1]. All faculty are required to have a singed SME on file with department.

Faculty input: Expectations are standard with respect to 3rd year Full Professors.

Based on my current research, service and teaching output, and departmental policies, my current research effort level corresponds to very active research, and my current service effort level corresponds to above base-line service. Based on this, my corresponding teaching effort for the next SME period[2] is 2 courses per academic year.

ii) Teaching (academic activities) – Teaching Responsibilities

Departmental expectations: Excellence in teaching is expected. Current departmental teaching load and other related policies will be followed. Teaching of both undergraduate and graduate courses, and mentoring and advising of undergraduate and graduate students is required. Leadership and participation in the design and implementation of new courses, and in the revision of existing core and area of specialty courses, is expected. Faculty are expected to pro-actively engage in all academic activities of import to the department.

Faculty input: My teaching effort for the next SME period is 2 courses per academic year.

I will be teaching the graduate software engineering (CSC 510) and automated software engineering (a 500-level, 700-level special topics course. I will officially advise between 4 and 8 graduate students, and 2 or fewer undergraduate research assistants. I will be serving on several exam and dissertation committees.

iii) Scholarship (research and innovation) – Research Areas

Departmental expectations: Excellence in scholarship is expected. An active, funded, peer-reviewed, nationally and internationally prominent research program in chosen areas of expertise is expected. This includes research, publications and direction of PhD and MS students to successful completion (as chair or co-chair). High-quality scholarship is expected to be a) funded well beyond the individual faculty salary level (including release time) over long periods of time, and b) is expected to support graduate students. Faculty are expected to engage pro-actively in all scholarship activities of import to the department.

Faculty input: During the next SME period I plan to have my research effort at the very active research level.

Research areas include software engineering and automated software engineering.

iv) Professional Activities (service)

Departmental expectations: Excellence in leadership and professional activities is expected. All faculty are expected to

[1] For example, during annual faculty evaluation period (spring semester every academic year), when a major RPT or PTR action occurs, or when status or activity level of the faculty member changes (e.g., sabbatical, leave of absence, partial/phased retirement, major changes in duties, etc.), and at least once within a PTR cycle of the faculty member (annually for assistant, every 3 years for associate, and every 5 years for full professors). Updates may be issued via email, or some other form of written communication, as addenda to the signed SME.

[2] New SME period typically covers at least one academic year into the future, however the document may cover longer periods if that is appropriate. Current (past) period performance, on which future teaching load expectations are determined, is based (barring special arrangements, such as start-up) on a moving average analysis of research, service and teaching performance described in the CSC Definitions document. Faculty teaching load, and other duties, are reviewed and updated by the department head on as needed basis, and at least once a year during the annual faculty review process.

participate in departmental, college and university level committees and other governance activities and roles. All faculty are expected to participate in relevant external professional activities (e.g., professional societies, conference program committees, national and international professional bodies and activities).

Faculty input: During the next SME period I plan to have my service effort at above base-line service level. I am currently serving on the Software Engineering Search Committee. Additionally, I will:

- Serve on the program committee for several major conferences and workshops, most notably ICSE'16, ASE'15, FSE'15, ISSRE'15.
- Develop new books on data science for software engineering.
- Continue to review papers from numerous top-level conferences.
- Take on such duties as required as part of being co-chair for ICSME'16
- I will lead the development of an NSF SE Research Experience for undergraduates grant.
- Lastly, I will administrate PROMISE, a repository for the empirical SE community

Prof. Tim Menzies

BRIEF RESUME

1. Education background:

- Ph.D., CS, University of New South Wales, 1995 *Generalized Testing of Knowledge Bases*; Advisor Paul Compton
- Masters of Cognitive Science, University of New South Wales, Australia, 1988
- B.S. Computer Science, University of New South Wales, 185.

2. Professional experience:

- August 2014 to present: Professor, CS, North Carolina State University, Raleigh, NC
- May 2012 to August 2014: Professor, West Virginia University, Morgantown, WV
- February 2006 to April 2012, Associate Professor, West Virginia University, Morgantown, WV
- December 2001 to December 2003, SE research chair, NASA IV&V Facility, West Virginia
- July 2000 to January 2001: Assistant professor, University of British Columbia, Vancouver, CA.
- June 1996 to June 1998: Vice- Chancellor's Research Fellow**,** University of New South Wales,
- February 1995 to June 1996: Assistant professor, Monash University, Australia

3. Scholarly and creative activities:

Books	*Career*	*Post Tenure*	*Current Year*
Authored books	0	-	-
Edited books and Proceedings	2	2	1
Refereed book chapters	13	3	-

Papers, Articles, Patents, Reports, etc.	*Career*	*Post Tenure*	*Current Year*	*Submitted*
Refereed journal articles	63	43	2	3
Refereed magazine articles	4	3	1	
Other magazine articles	-	-	-	
Refereed conference papers	107	45	2	3
Refereed workshop papers	68	13	1	
Refereed panel papers	-	-	-	
Refereed posters/fast abstract	-	-	-	
Technical reports	4	-	-	
Refereed tutorials	4	4	2	
Course pack (with ISBN)	-	-	-	-
News interviews	4	2	-	-

Talks, Presentations	*Career*	*Post Tenure*	*Current Year*
Keynotes and distinguished speaker	7	4	2
Other invited talks	-	-	-

Funded Research, Development and Teaching	*Career*	*Post Tenure*	*Current Year*
Contracts and Grants	$7,864,702	$6,025,542	$602,122
Gifts (cash)	-	$260,000	$210,000
Gifts (in kind)	-	-	-
Other: PhD Fellowships	-	-	-

Mentoring and Supervision (see CV for details)	*Career*	*Post Tenure*	*Current Year*
PhD (chair/co-chair), graduated	7	6	-
PhD (chair/co-chair), current	-	-	9
MS (chair/co-chair), graduated	27	20	2
MS (chair/co-chair), current	-	-	2
Undergraduate advisees, graduated	4	2	2
Faculty mentored	-	3	3

Courses taught	*Career*	*Post Tenure*	*Current Year*
Regular undergraduate (3 credits, 10< x < 100 students)	8	7	-
Large undergraduate (3 credits, x > 100 students)	5	-	-
Regular graduate (3 credits, 10 < x < 100 students)	28	18	2

Courses created and/or revised in a significant way	*Career*	*Post Tenure*	*Current Year*
Undergraduate	4	2	-
Graduate	6	3	1

Other	*Career*	*Post Tenure*	*Current Year*
Development of Software Packages	5	2	10
Creation/Direction of Dept. Facilities – Labs & Centers	2	-	-
Major awards and recognitions	6	6	3
Major off-campus services	-	-	-

4. Membership in professional organizations:
- Association for Computing Machinery (ACM), 1996-present
- Institute of Electrical and Electronic Engineers (IEEE), 1997-present

5. Scholarly and professional honors:
- **Distinguished reviewer, ACM Transactions on Software Engineering Methodologies, 2016**
- **Outstanding reviewer award, journal of Information and Software Technology, 2016**
- **Service award from Big Data community: Lexis Nexis, 2015**
- Distinguished reviewer, ACM Transactions on SE Methodologies, 2015
- WVU College of Engineering, Outstanding Researcher, 2010
- NASA Commendation for Chief of Mission Assurance, 2004

6. Professional service on campus:
- **NC State Member , CSC Faculty Search (2015, 2016)**
- NC State Member , Software Engineering Faculty Search (2014)
- NC State, Open house weekend (March 2015)
- Curating the PROMISE repository of SE data
- WVU, computer science, Promotion & Tenure committee (2010-2014)
- WVU, Member, Faculty Search Committees (2010-2013)
- Director, National Archives/WVU project (2009-2011)
- Director, WVU/NASA Research Collaboration (2002-2009)

7. Professional service off campus (see CV for complete list):
- **Co-General Chair: International Conference on Software Maintenance and Evolution 2016**
- **Co-Program Chair: SSBSE'16**, ICSE NIER'15, ASE'12.
- **Editorial Board: Information Software Technology**
- **Editorial Board: Software Quality Journal**
- **Editorial Board: Big Data Research Journal**.
- Associate Editor: IEEE Transactions on Software Engineering 2011-present.
- Editorial Board: Empirical Software Engineering International Journal, 2009-present; Automated Software Engineering Journal, 2010-present
- Steering Committee Member: IEEE Conference on Automated Software Engineering (2012-present); PROMISE conference 2005-2012.
- PC member: **ASE'16, BIGDSE'16, EASE 2016, ESEM2016, ICSE-SRC 2016, ISSRE 2016, PROMISE 2016, RAISE 2016, SCORE 2016**, Icse'15, Ase'15, BigDSE'15, Ease'15, EsPreSSE'15, Esem'15, Fse'15, Gecco'15, Icpc'15, Issre'15, Msr'15, NasBase'15, Promise'15, Raise'15, Ssbse'15; Previously: MSR'14, ICSE14-demos, ICSE14-mainConference, DAPSE'14, EASE'14, GTSE'14, SAM 2014, SEAA 2014, MSR (2011-2014). ASE (2002-20114), ESEM (2011-2013) • SAM2103, DAPSE'13, ICSE'13: demos ,ASE-Tools'13 , ISSRE'13, GTSE'13, MALIR'13 , Software Mining -2012, 2013 , ISSRE'09, ISSRE'10 and many more dating back to 1991.

II. TEACHING AND MENTORING OF UNDERGRADUATE AND GRADUATE STUDENTS

A. TEACHING EFFECTIVENESS

Course	When	Responses/Enrolled	Explanation	Effectiveness	Knowledge	Excellence
CSC 791/791 (ASE)	Fall 2015	44 / 52	4.3 / 4.2*	4.4 / 4.3*	4.5 / 4.4*	4.4 / 4.2*
CSC 510	Spring 2015	25 / 32	4.2 / 4.2*	4.4 / 4.3*	4.3 / 4.4*	4.3 / 4.2*
CSC 791 (ASE)	Fall 2014	5 / 5	5.0 / 4.0*	5.0 / 4.0*	5.0 / 4.3*	5.0 / 4.0*

* departmental average

Student comments:

- Prof. Warwick Arden (Provost) mailed me on Mark 16 2016 saying that some students had used the "Thank a Professor" website to comment favorable on my teaching:
  - He writes "You efforts with these students area reflection of your dedication to teaching and learning. I congratulate you on this recognition and offer by sincere appreciate for your work with students at NCSU.
  - Student comments included "I compare you to an owl because you have a deep connection with wisdom and intuitive knowledge. You have been kind and patient with me and your students. Thank you for everything."
- **CSC510**:
  - Enthusiastic, Very passionate … Dr. Menzies is a fantastic teacher. He loves research work and hard-working. … Excellent teacher with a flair for creating interests in students …. instructors shares his own experience which is valuable for students and gives pretty god idea who software industry works. … Assignments were good and helped in learning. … Tim is a very good professor, whose class is full of fun. He always told something new to us. … Awesome professor … Amazing professor. His enthusiasm for the subject could be felt. … Amazing professor. His enthusiasm for the subject could be felt.
- **Csc591/791**:
  - Dr. Menzies is overall an incredibly reasonable instructor that presents material in a way that students can understand it and his expectations can be met with adequate effort. Above all his expectations are well explained at the outset of the course and if you want to be successful in the class you can be successful. ….Professor Menzies was enthusiastic, humorous, and communicated well. He used helpful visualizations, often an upside-down table or contorted elbow, as an analogue for difficult concepts…. I think the instructor is quite enthusiastic and energetic. He makes the class very much interactive…. Very enthusiastic and very intelligent. Conveys the ideas well and uses "perfect" examples …. Using the right example can save a lot of time which can be used to do more. Course was really good. Content was nice and its very relevant too. Further professor had recommended a book to follow for those who had not done much in python. He recommended it much before the classes started. I finished the book before the classes started and it helped me to become very comfortable with python as the course started. … brilliant course … The Instructor was well prepared and very enthusiastic…. The Instructor was well prepared and very enthusiastic…. Class is *never* boring; even if I had a hard time following the lectures at times, I walked away with new thoughts each class session…. Dr. Menzies is blunt with feedback but it's never personal….. Definitely the most difficult subject I've taken so far. I learned a lot and would do it again if given the choice….. This is cutting edge stuff and is worth the trouble; it's a very different course from a lot of the current offerings, however, and I would highly recommend for any prospective or current PhD students interested in research.

Also, I have taught the following other courses (at previous universities):

- Programming languages , (2009, 2010, 2011, 2012,2013, 2014), 3rd year undergraduate subject
- AI , 2011,2012,2013, 2014 4th year undergraduate subject
- Data mining, (2002,2003,2004,2006,2007,2008,2009,2010,2011,2012,2013) graduate subject
- AI (2008, 2009,2010,2011), graduate subject
- Search-based software engineering (2009,2012, 2014), graduate subject
- Agent-oriented programming (2009), Ph.D. graduate subject
- Open Source Software (2007), 4th year undergraduate subject
- Lightweight Software Engineering (2004), 4th year undergraduate subject
- Knowledge engineering (2002, 2003), 4th year undergraduate subject
- Software V&V (2003), Masters course year
- Modelling and analysis of software (2000), 4th year undergraduate subject
- Domain specific languages (2001), graduate class.
- OO software development (1997-98), 4th year undergraduate subject
- Visual programming (1996), 3rd year undergraduate subject
- Software engineering (1996), 3rd year undergraduate subject
- Research methods (1995,1996), 4th year undergraduate subject

B. INSTRUCTIONAL DEVELOPMENT

- ***New course Automated Software Engineering:*** **At NC State in 2015 I created and taught a new subject automated software engineering. All lectures and projects were written by me.**
- *New course Search-based Software Engineering:* At NC State in 2014 I created and taught a new subject on search-based software engineering. An updated version of this will be taught in Fall 2015 as Automated (model-based) Software Engineering.
- *New course Artificial Intelligence.* At West Virginia University in 2008, 2010, 2012 I created then completely updated an undergraduate subject on artificial intelligence.
- *Programming languages*. At West Virginia University in 2009 and 2011 I updated the undergraduate programming language subject to include logic programming and functional programming.
- *Data mining:* At West Virginia University in 2002, I created and taught a graduate subject on this topic. This subject was extensively revised each year 2003 to 2013.
- *Agent*-oriented programming: At West Virginia University in 2009, I created and taught a graduate subject on this topic.
- *Knowledge engineering:* At West Virginia University in 2002, I created and taught a graduate subject on this topic.
- *Domain specific languages.* At the University of British Columbia in 2001, I created and taught a graduate subject on this topic.
- *Visual programming languages.* At Monash University in 1996, I created and taught a graduate subject on this topic.
- *Research methods.* At Monash University in 1995, I created a subject on graduate research methods.

C. MENTORING ACTIVITIES

- **Working with my SE faculty colleagues, developed a successful NSF Research Experience for Undergraduates grant for "Science of Software", awarded 2016.**
- **Mentor for**
  - **Associate professor Dennis Bahler**
  - **Associate professor Vince Freeh (meet, several times)**
  - **Assistant professor Kathryn Stolee (meet, numerous times)**

D. MASTERS AND DOCTORAL THESES DIRECTED

I am or was chair or co-chair of the advisory committee for the following research students by thesis (students who have graduated= 7 PhD + 27 MS):

Student working towards a degree (9 PhD):

1. **Vivek Nair (passed written preliminary, April 2016)**
2. **Wei Fu (passed written preliminary, April 2016)**
3. **Rahul Krishna**
4. **George Mathew**
5. **Di Chen (Jack)**
6. **Mahmoud Abdel-Fattah**
7. **Zhe Yu**
8. **Rahul Krishna**
9. **Jianfeng Chen**

Completed Ph.D.:

1. Abdel Sayyad Ph.D. (2014, WVU) *Evolutionary Search Techniques with Strong Heuristics for Multi-Objective Feature Selection in Software Product Lines*
2. Joe Krall Ph.D. (2014, WVU) *Active Learning for Search-Based Software Engineering*
3. Fayola Peters Ph.D. (2014, WVU) *Privacy and Data Sharing*
4. Ekrem Ph.D. (2012, WVU) *A Principled Methodology: A Dozen Principles of Software Effort Estimation*
5. Nandeshwar, Ashutosh Ph.D. (2011, WVU) *Longitudinal study of first-time freshmen using data mining*
6. David Owen Ph.D. (2010, WVU) *Combining complementary formal verification strategies to improve performance and accuracy*
7. Scott Chen Ph.D. (2004, U.Sc.) *Data Mining for Effort Estimation*

Completed Masters:

1. **George Mathew, MS, NC State**
2. **Rahul Krishna, MS, NC State**

3. Divya Ganesan MS (2015,WVU) Exploring Essential Content of Defect Prediction and Effort Estimation through Data Reduction
4. Ben Province MS (2015, WVU), *The Effects of Parameter Tuning on Machine Learning Performance in a Software Defect Prediction Context.*
5. Vasil Papakroni MS (2013, WVU) *Data Carving: Identifying and Removing Irrelevancies*
6. Joseph Craig MS (2013, WVU) *Accelerating MOEA Non-dominated Sorting by Preserving Archival Relationships*
   Will Burney MS (2012, WVU) *Understanding Search-Based Software Engineering*
7. Adam Brady MS (2011, WVU) *W2 : a simple, flexible, case-based recommendation engine for software quality*
8. Brian Lemon MS (2010, WVU) *The effect of locality based learning on software defect prediction*
9. Fayola Peters MS (2010, WVU) *CLIFF: finding prototypes for nearest neighbor algorithms with applications*
10. Andrew Matheny MS (2010, WVU) *Trade-offs of heuristic vs. rigorous algorithms in text mining*
11. Joe D'alessandro MS (2010, WVU) *Optimized trusted information sharing*
12. Grey Gay MS (2010, WVU) *The robust optimization of non-linear requirements models*
13. Adam Nelson MS (2010, WVU*) OURMINE: an open source data mining toolkit*
14. Ous El-waras MS (2008, WVU) *Software process control without calibration*
15. Omid Jalali MS (2008, WVU) *Evaluation bias in effort estimation*
16. Zach Milton MS (2008, WVU) *Which: a stochastic best-first search learner*
17. Brian Sower MS (2008, WVU) *Increasing the performance and realism of procedurally generated buildings*
18. Justin DiStefano MS (2008, WVU) *Building better software : the applicability of a professional tool for automa*
19. Daniel Baker MS (2007, WVU) *Hybrid approach to expert and model based effort estimation*
20. Donald Boland MS (2007, WVU) *Data discretization simplified: randomized binary search trees for data preprocessing*
21. Jeremy Greenwald MS (2006, Portland State) *Understanding procedural Knowledge*
22. Ryan Clark MS (2005, Portland State) *Optimizing Treatment Learning*
23. Kareem Ammar (2004, WVU) *Multi-heuristic theory assessment with iterative selection* [
24. Yi Hu MS (2003, University British Columbia) Treatment learning
25. Eliza Chaing MS (2003, University British Columbia) *Early LifeCycle Simulation of Software Process Models*.
26. David Owen MS (2002,WVU) *Combining complementary formal verification strategies to improve performance and accuracy*
27. John Powell MS (2001,WVU) *Graph theoretic approach to assessing tradeoffs on memory usage for model che*cking

III. SCHOLARSHIP IN THE REALMS OF FACULTY RESPONSIBILITY

A. SCHOLARLY ACCOMPLISHMENTS – PUBLICATIONS

Submitted (under review)

2 * EMSE journal (with CSC Ph.D. student George Mathew)
1 * Information Software Technology (with CSC Ph.D, student Wei Fu)
1 * Conference on Foundations of Software Engineering (with CSC Ph.D. student Rahul Krishna)
2 * Conference on search-based SE (with CSC Ph.D. students Jian Feng Chen and Rahul Krishna)

Invited and Contributed Research Presentations

1. **Keynote, ICSE'16 workshop keynote (SBST'16): Testing: the (w)hole story.**
2. **Tutorial, ICSE'16: How Not to do it: Anti-patterns in data science**
3. Keynote, ICSE'15 workshop keynote (WetSOM'14): What Metrics matter. Hyderabad, India.
4. Tutorial, ICSE'15: Art and Science of Analyzing Software Data

Refereed Journal and Top Magazine Publications

1. **W. Fu, T Menzies, X. Shen, Tuning for Software Analytics: is it Really Necessary? Journal of Information and Software Technology, accepted for publication 2016.**
2. **J. Krall, T. Menzies and M. Davies, "GALE: Geometric Active Learning for Search-Based Software Engineering," in IEEE Transactions on Software Engineering, vol. 41, no. 10, pp. 1001-1018, Oct. 1 2015.**
3. Krall J., Menzies T., Davis,M. Better Model-Based Analysis of Human Factors for Safe Aircraft Approach, submitted, IEEE Transactions on Human Machine System, accepted with minor revision Feb 2014
4. Transfer learning in effort estimation, E Kocaguneli, T Menzies, E Mendes Empirical Software Engineering, 1-31, 2014
5. SN Partington, V Papakroni, T Menzies , Optimizing data collection for public health decisions: a data mining approach, BMC Public Health 14 (1), 593, 2014
6. Reduced Item Food Audits based on the Nutrition Environment Measures Surveys, Susan Partington, Glanz, Karen, Saelens, Brian, Colburn, Trina, Menzies, Tim. American Journal of Preventive Medicine., accepted, to appear
7. The International Center of Excellence in Software Engineering: Accomplishments and Challenges, Shata, M Salah Hamdi, W Abdelmoez, T Menzies, H Ammar, Communications of the ACS 6 (2), 4-11,2014
8. Incremental Development of Fault Prediction Models Yue Jiang, Bojan Cukic, Tim Menzies, Jie Lin, International Journal of Software Engineering and Knowledge Engineering, 23(10), p1399-1425, 2013
9. Ekrem Kocaguneli, Tim Menzies: Software effort models should be assessed via leave-one-out validation. Journal of Systems and Software 86(7): 1879-1890 (2013)
10. Fayola Peters, Tim Menzies, Liang Gong, Hongyu Zhang: Balancing Privacy and Utility in Cross-Company Defect Prediction. IEEE Trans. Software Eng. 39(8): 1054-1068 (2013)
11. Learning Project Management Decisions: A Case Study with Case-Based Reasoning Versus Data Farming T Menzies, A Brady, J Keung, J Hihn, S Williams, O El-Rawas, P Green, , Barry Boehm, IEEE Transactions on Software Engineering, 39(12), oo1698-1713, 2013
12. Ekrem Kocaguneli, Tim Menzies, Jacky W. Keung: Kernel methods for software effort estimation - Effects of different kernel functions and bandwidths on estimation accuracy. Empirical Software Engineering 18(1): 1-24 (2013)
13. "Local vs. Global Lessons for Defect Prediction and Effort Estimation" by Tim Menzies, Andrew Butcher, David Cok, Andrian Marcus, Lucas Layman, Forrest Shull, Burak Turhan, Thomas Zimmermann, IEEE Transactions on Software Engineering, 2013
14. Kocaguneli, E.; Menzies, T.; Keung, J.; Cok, D.; Madachy, R.; , "Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data," Software Engineering, IEEE Transactions on ,
15. Jacky Keung, Kocaguneli, Ekrem, Menzies, Tim , "Finding conclusion stability for selecting the best effort predictor in software effort estimation" , Automated Software Engineering, p1-25, May 2012,
16. Markus Lumpe, Rajesh Vasa, Tim Menzies, Rebecca Rush, Burak Turhan:Learning Better Inspection Optimization Policies. International Journal of Software Engineering and Knowledge Engineering 22(5): 621-644 (2012)
17. Ekrem Kocaguneli, Tim Menzies, Ayse Bener, Jacky W. Keung: Exploiting the Essential Assumptions of Analogy-Based Effort Estimation. IEEE Trans. Software Eng. 38(2): 425-438 (2012)
18. "On the Value of Ensemble Effort Estimation" by E. Kocaguneli and Tim Menzies and J. Keung. IEEE Transactions on Software Engineering,2011 . 38(6): 1403-1416 (2012)

19. J. Krall and T.J. Menzies, “Aspects of Replayability and Software Engineering: Towards a Methodology of Developing Games” Journal of Software Engineering and Applications 5 (7), 459-466, 2012
20. H. H. Ammar and T. Menzies and O. Shata and A. Erradiand M. Kessentini and W. Abdelmoez and , M. Kholief and M. Shaheen and M. Abdelhamid, and A AbdelHamid and M.A. Omar and Mohamed Salah Hamdi. “The International Center of Excellence in Software Engineering” Communications of the Arab Computer Society, Vol. 4 No.2, December, 2011
21. Exploring the Effort of General Software Project Activities with Data Mining" by Topi Haapio and Tim Menzies. International Journal of Software Engineering and Knowledge Engineering pages 725-753 2011.
22. "Learning patterns of university student retention" by Ashutosh Nandeshwar and Tim Menzies and Adam Nelson. Expert Systems with Applications , volume 38, number 12, pages 14984 – 14996, 2011 .
23. "What is Enough Quality for Data Repositories?" by Tim Menzies and Adam Brady and Ekrem Kocaguneli. Software Quality Professional, volume 13, number 3, 2011 .
24. A. Tosun and A. Bener and B. Turhan and T. Menzies, "Practical considerations in deploying statistical methods for defect prediction: A case study within the Turkish telecommunications industry" by Information and Software Technology pages 1242-1257 2010 . Available from http://menzies.us/pdf/10practical.pdf .
25. T.J. Menzies and Z. Milton and B. Turhan and B. Cukic and Y. Jiang and A. Bener , "Defect Prediction from Static Code Features: Current Results, Limitations, New Approaches" in Automated Software Engineering December 2010 . Available from http://menzies.us/pdf/10which.pdf .
26. Adam Nelson, Tim Menzies, Gregory Gay, "Sharing Experiments Using Open Source Software" in Software-Practice and Experience September 2010 . Available from http://menzies.us/pdf/10ourmine.pdf .
27. Tim Menzies and Omid Jalali and Jairus Hihn and Dan Baker and Karen Lum, "Stable Rankings for Different Effort Models" by. Automated Software Engineering December 2010 . Available from http://menzies.us/pdf/10stable.pdf .
28. Adam Brady and Tim Menzies and Oussama El-Rawas and Ekrem Kocaguneli and Jacky Keung, "Case-Based Reasoning for Reducing Software Development Effort" in Journal of Software Engineering and Applications 2010 . Available from http://menzies.us/pdf/10w0.pdf .
29. Oussma El-Rawas and Tim Menzies, "A Second Look at Faster, Better, Cheaper" in.Innovations Systems and Software Engineering pages 319-335 2010 .
30. Gregory Gay and Tim Menzies and Misty Davies and Karen Gundy-Burlet, "Automatically finding the control variables for complex system behaviour" in Automated Software Engineering December 2010 . Available from http://menzies.us/pdf/10tar34.pdf. .
31. James H. Andrews and Tim Menzies and Felix Li , "Genetic Algorithms for Randomized Unit Testing" in IEEE Transactions on Software Engineering March 2010 . Available from http://menzies.us/pdf/10nighthawk.pdf .
32. T. Menzies and S. Williams and O. Elrawas and D. Baker and B. Boehm and J. Hihn and K. Lum and R. Madachy, “Accurate Estimates Without Local Data?" Software Process Improvement and Practice pages 213-225 July 2009 . Available from http://menzies.us/pdf/09nodata.pdf .
33. G. Gay and T. Menzies and O. Jalali and G. Mundy and B. Gilkerson and M. Feather and J. Kiper, "Finding robust solutions in requirements models" , Automated Software Engineering December 2009 . Available from http://menzies.us/pdf/09keys2.pdf
34. T. Menzies and O. Mizuno and Y. Takagi and Y. Kikuno, "Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects" by Journal of Software Engineering and Applications pages 221-236 November 2009.
35. B. Turhan, T. Menzies, A. Bener, and J. Distefano. On the relative value of cross-company and within-company data for defect prediction. Empirical Software Engineering, 2009. Available from http://menzies.us/pdf/08ccwc.pdf.
36. T. Menzies, M. Benson, K. Costello, C. Moats, M. Northey, and J. Richarson. Learning better IV&V practices. Innovations in Systems and Software Engineering, March 2008. Available from http://menzies.us/pdf/07ivv.pdf.
37. M. Feather, S. Cornford, K Hicks, J. Kiper, and T. Menzies. Application of a broad-spectrum quantitative requirements model to early-lifecycle decision making. IEEE Software, May 2008. Available from http://menzies.us/pdf/08ddp.pdf.
38. Tim Menzies, Jeremy Greenwald, and Art Frank. Data mining static code attributes to learn defect predictors. IEEE Transactions on Software Engineering, January 2007. Available from http: //menzies.us/pdf/06learnPredict.pdf.
39. Tim Menzies, Alex Dekhtyar, Justin Distefano, and Jeremy Greenwald. Problems with precision. IEEE Transactions on Software Engineering, September 2007. http://menzies.us/pdf/ 07precision.pdf.
40. T. Menzies and Y. Hu. Just enough learning (of association rules): The TAR2 treatment learner. In Artificial Intelligence Review, 2007. Available from http://menzies.us/pdf/07tar2. pdf.
41. T. Menzies, D.Owen, and J. Richardson. The strangest thing about software. IEEE Computer, 2007. http://menzies.us/pdf/07strange.pdf.
42. Tim Menzies, Zhihao Chen, Jairus Hihn, and Karen Lum. Selecting best practices for effort estimation. IEEE Transactions on Software Engineering, November 2006. Available from http://menzies.us/pdf/06coseekmo.pdf.
43. T. Menzies and J. Richardson. Making sense of requirements, sooner. IEEE Computer, October 2006. Available from

http://menzies.us/pdf/06qrre.pdf.

scorecT. Menzies and J. Hihn. Evidence-based cost estimation for better quality software. IEEE Software, July/August 2006. Available on-line at http://menzies.us/pdf/06costs.pdf.

44. T. Menzies and C. Pecheur. Verification and Validation and Artificial Intelligence. In M. Zelkowtiz, editor, Advances in Computing, volume 65. Elsevier, 2005. Available from http://menzies. us/pdf/04aivv.pdf.
45. T. Menzies, R. Gunnalan, K. Appukutty, Srinivasan A, and Y. Hu. Learning tiny theories. In International Journal on Artificial Intelligence Tools (IJAIT), to appear, 2005. Available from http://menzies.us/pdf/03select.pdf.
46. Zhihao Chen, Tim Menzies, Dan Port, and Barry Boehm. Finding the right data for software cost modeling. IEEE Software, Nov 2005.
47. T.J. Menzies, R.F. Cohen, S. Waugh, and S. Goss. Applications of abduction: Testing very long qualitative simulations. IEEE Transactions of Data and Knowledge Engineering, pages 1362– 1375, November/December 2003. Available from http://menzies.us/pdf/97iedge. pdf.
48. T. Menzies and J.S. Di Stefano. More success and failure factors in software reuse. IEEE Transactions on Software Engineering, May 2003. Available from http://menzies.us/pdf/ 02sereuse.pdf.
49. T. Menzies and Y. Hu. Data mining for very busy people. In IEEE Computer, November 2003. Available from http://menzies.us/pdf/03tar2.pdf.
50. E. Chiang and T. Menzies. Simulations for very early lifecycle quality evaluations. Software Process: Improvement and Practice, 7(3-4):141–159, 2003. Available from http://menzies. us/pdf/03spip.pdf.
51. T. Menzies and B. Cukic. When to test less. IEEE Software, 17(5):107–112, 2000. Available from http://menzies.us/pdf/00iesoft.pdf.
52. T. Menzies and B. Cukic. Adequacy of limited testing for knowledge based systems. International Journal on Artificial Intelligence Tools (IJAIT), June 2000. Available from http://menzies. us/pdf/00ijait.pdf.
53. T. Menzies, K.D. Althoff, Y. Kalfoglou, and E. Motta. Issues with meta-knowledge. International Journal of Software Engineering and Knowledge Engineering, 10(4), August 2000. Available from http://menzies.us/pdf/00sekej.pdf.
54. Y. Kalfoglou, T. Menzies, K.F. Althoff, and E. Motta. Meta-knowledges in systems design: panacea... or undelivered promise? The Knowledge Engineering Review, 15(4), December 2000. Available from http://menzies.us/pdf/00ker.pdf.
55. Tim Menzies. Critical success metrics: Evaluation at the business-level. International Journal of Human-Computer Studies, special issue on evaluation of KE techniques, 51(4):783–799, October 1999. Available from http://menzies.us/pdf/99csm.pdf.
56. T. Menzies. Knowledge maintenance: The state of the art. The Knowledge Engineering Review, 14(1):1–46, 1999. Available from http://menzies.us/pdf/97kmall.pdf.
57. T. Menzies. Cost benefits of ontologies. ACM SIGART Intelligence magazine, Fall 1999. Available from http://menzies.us/pdf/99sigart.pdf.
58. T.J. Menzies. Towards situated knowledge acquisition. International Journal of Human-Computer Studies, 49:867–893, 1998. Available from http://menzies.us/pdf/98ijhcs.pdf.
59. T.J. Menzies and P. Compton. Applications of abduction: Hypothesis testing of neuroendocrinological qualitative compartmental models. Artificial Intelligence in Medicine, 10:145–175, 1997. Available from http://menzies.us/pdf/96aim.pdf.
60. T.J. Menzies. OO patterns: Lessons from expert systems. Software Practice and Experience, 27(12):1457–1478, December 1997. Available from http://menzies.us/pdf/ 97patern.pdf.
61. T.J. Menzies. Applications of abduction: Knowledge level modeling. International Journal of Human Computer Studies, 45:305–355, 1996. Available from http://menzies.us/pdf/ 96abkl.pdf.
62. T.J. Menzies. An investigation of the ai and expert systems literature 1980-1984. AI Magazine, Summer 1989.
63. T.J. Menzies. Domain-specific knowledge representations. AI Expert, Summer 1989.

Books/Book Chapters

1. **The Art and Science of Analyzing Software Data, C. Bird, T. Menzies, T. Zimmermann, Morgan Kaufmann, 2016**
2. Sharing Data and Models in Software Engineering, T. Menzies, Ekrem Kocaguneli, L. Minku, F. Peters, B. Turhan, Morgan Kaufmann, 2014
3. Occam's Razor and Simple Software Project Management T Menzies Software Project Management in a Changing World, 447-472, 2014
4. Data mining: a tutorial T Menzies, Recommendation Systems in Software Engineering. Springer, Berlin, 2014
5. "The Quest for Convincing Evidence" by Tim Menzies and Forrest Shull. Making Software: What Really Works and We We Believe it 2010

6. "Condensing uncertainty via Incremental Treatment Learning" by T. Menzies and E. Chiang and M. Feather and Y. Hu and J.D. Kiper. Software Engineering with Computational Intelligence 2003 . Available from http://menzies.us/pdf/02itar2.pdf .
7. "Many Maybes Mean (Mostly) the Same Thing" by T. Menzies and H. Singh. Soft Computing in Software Engineering 2003 . Available fromhttp://menzies.us/pdf/03maybe.pdf .
8. "How Many Tests are Enough?" by T.J. Menzies and B. Cukic. Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available fromhttp://menzies.us/pdf/00ntests.pdf .
9. "SE/KE Reuse Research: Common Themes and Empirical Results" by T.J. Menzies.Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available from http://menzies.us/pdf/00reuse.pdf
10. "Knowledge Elicitation: the State of the Art" by T.J. Menzies. Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available fromhttp://menzies.us/pdf/00getknow.pdf .
11. "Evaluation Issues for Visual Programming Languages" by T. Menzies. Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available fromhttp://menzies.us/pdf/00vp.pdf
12. "Practical Machine Learning for Software Engineering and Knowledge Engineering" by T. Menzies. Handbook of Software Engineering and Knowledge Engineering December 2001 . Available from http://menzies.us/pdf/00ml.pdf .
13. "Expert Systems Maintenance" by T.J. Menzies and J. Debenham. Encyclopedia of Computer Science and Technology pages 35-54 2000 . Available fromhttp://menzies.us/pdf/00cst.pdf .
14. "Software Visualization" by P. Haynes and T. Menzies and R.F. Cohen. 1997 . Available from http://menzies.us/pdf/oovis95.pdf .

Refereed Conference Publications

1. **Lucas Layman, Allen Nikora, Joshua Meek, TIm Menzies,, Topic Modeling NASA Space System Problem Reports (research in Practice Track) , MSR'16 (27% acceptance rate for full papers)**
2. **Jairus Hihn, Tim Menzies, Improving and Expanding NASA Software Cost Estimation Methods 2016 IEEE Aerospace Conference.**
3. Scalable product line configuration: A straw to break the camel's back, ASE , 2013 , AS Sayyad, J Ingram, T Menzies, H Ammar
4. Abdel Salam Sayyad, Tim Menzies, Hany Ammar: On the value of user preferences in search-based software engineering: a case study in software product lines. ICSE 2013: 492-501
5. Class level fault prediction using software clustering, G Scanniello, C Gravino, A Marcus, T Menzies, ASE 2013
6. Sonia Haiduc, Gabriele Bavota, Andrian Marcus, Rocco Oliveto, Andrea De Lucia, Tim Menzies: Automatic query reformulations for text retrieval in software engineering. ICSE 2013: 842-851
7. Tim Menzies: Beyond data mining; towards "idea engineering". PROMISE 2013: 11Learning from open-source projects: An empirical study on defect prediction, Z He, F Peters, T Menzies, Y Yang, ESEM 2013
8. Ekrem Kocaguneli, Bojan Cukic, Tim Menzies, Huihua Lu: Building a second opinion: learning cross-company data. PROMISE 2013: 12
9. Beyond data mining; towards idea engineering T Menzies, PROMISE 2013
10. Ekrem Kocaguneli, Thomas Zimmermann, Christian Bird, Nachiappan Nagappan, Tim Menzies: Distributed development considered harmful? ICSE 2013: 882-890
11. Fayola Peters, Tim Menzies: Privacy and utility for defect prediction: Experiments with MORPH. ICSE 2012: 189-199
12. Yang Sok Kim, Byeong Ho Kang, Seung Hwan Ryu, Paul Compton, Soyeon Caren Han, Tim Menzies: Crowd-Sourced Knowledge Bases. PKAW 2012: 258-271
13. Raymond Borges, Tim Menzies: Learning to change projects. PROMISE 2012: 11-18
14. Ekrem Kocaguneli, Tim Menzies, Jairus Hihn, Byeong Ho Kang: Size doesn't matter?: on the value of software size features for effort estimation. PROMISE 2012: 89-98
15. "How to Find Relevant Data for Effort Estimation?" by Kocaguneli, E. and Menzies, T..Proceedings ESEM11, 2011
16. "Local vs Global Models for Effort Estimation and Defect Prediction" by Menzies, Tim and Butcher, Andrew and Marcus, Andrian and Zimmermann, Thomas and Cok, David. IEEE ASE11 2011 . Available from http://menzies.us/pdf/11ase.pdf .
17. "Text mining in supporting software systems risk assurance" by Huang, LiGuo and Port, Daniel and Wang, Liang and Xie, Tao and Menzies, Tim. IEEE ASE10 pages 163--166 2010. Available from http://menzies.us/pdf/10textrisk.pdf .
18. "On the Shoulders of Giants" by E. Barr and C. Bird and E. Hyatt and T. Menzies and G. Robles. FoSER 2010 November 2010 . Available from http://menzies.us/pdf/10giants.pdf.
19. "Case-Based Reasoning vs Parametric Models for Software Quality optimization" by Adam Brady and Tim Menzies. PROMISE10 2010 . Available from http://menzies.us/pdf/10cbr.pdf .
20. "Software is Data Too" by A. Marcus and T. Menzies. FoSER 2010 November 2010 . Available from http://menzies.us/pdf/10softwareisdata.pdf .

21. "When to Use Data from Other Projects for Effort Estimation" by Ekrem Kocaguneli and Gregory Gay and Tim Menzies and Ye Yang and Jacky W. Keung. IEEE ASE10 2010 . Available from http://menzies.us/pdf/10other.pdf .
22. "Regularities in Learning Defect Predictors" by Burak Turhan, Ayse Bener and Tim Menzies. Profes 2010 2010 . .
23. "On the Value of Learning From Defect Dense Components for Software Defect Prediction Proceedings of PROMISE10" by Hongyu Zhang and Adam Nelson and Tim Menzies. 2010 . Available from http://menzies.us/pdf/10dense.pdf .
24. P. Green and T. Menzies and S. Williams and O. El-waras, "Understanding the Value of Software Engineering Technologies" by IEEE ASE09 2009 . Available from http://menzies.us/pdf/09value.pdf.
25. T. Menzies and O. El-Rawas and J. Hihn and B. Boehm, "Can We Build Software Faster and Better and Cheaper?" by PROMISE09 2009 . Available fromhttp://menzies.us/pdf/09bfc.pdf .
26. K. Gundy-Burlet and J. Schumann and T. Menzies and T. Barrett, "Parametric Analysis of a Hover Test Vehicle Using Advanced Test Generation and Data Analysis" by AIAA Aerospace, 2009,
27. T. Menzies and S. Williams and O. El-rawas and B. Boehm and J. Hihn, "How to Avoid Drastic Software Process Change (using Stochastic Statbility)" by ICSE09 2009 . Available from http://menzies.us/pdf/08drastic.pdf .
28. G. Gay and S. Haiduc and A. Marcus and T. Menzies, "On the use of Relevance Feedback in IR-based Concept Location" by . IEEE ICSM09 2009 . Available fromhttp://menzies.us/pdf/09irrf.pdf .
29. B. Lemon and A. Riesbeck and T. Menzies and J. Price and J DAlessandro and R. Carlsson and T. Prifiti and F. Peters and H. Lu and D. Port. "Applications of Simulation and AI Search: Assessing the Relative Merits of Agile vs Traditional Software Development" IEEE ASE09 2009 . Available from http://menzies.us/pdf/09pom2.pdf .
30. Jamie Andrews and Tim Menzies, "On the Value of Combining Feature Subset Selection with Genetic Algorithms: Faster Learning of Coverage Models" PROMISE09 2009 . Available from http://menzies.us/pdf/09fssga.pdf .
31. G. Gay and T. Menzies and B. Cukic and Burak Turhan, "How to Build Repeatable Experiments" by PROMISE09 2009 . Available from http://menzies.us/pdf/09ourmine.pdf .
32. B. Cukic and T. Menzies and Y. Jiang, "Variance analysis in software fault prediction models" IEEE ISSRE09 2009 . Available from http://menzies.us/pdf/09irrf.pdf .
33. B. Cukic Y. Jiang and T. Menzies. Cost curve evaluation of fault prediction models. In Proceedings, ISSRE'08, 2008. Available from http://menzies.us/pdf/08costcurves.pdf.
34. D. Port, A. Olkov, and T. Menzies. Using simulation to investigate requirements prioritization strategies. In IEEE ASE'08, 2008. Available from http://menzies.us/pdf/ 08simrequire.pdf.
35. T. Menzies and A. Marcus. Automated severity assessment of software defect reports. In ICSM'08, 2008. Available from http://menzies.us/pdf/08severis.pdf.
36. T. Menzies, O. Elrawas, B. Barry, R. Madachy, J. Hihn, D. Baker, and K. Lum. Accurate estimates without calibration. In International Conference on Software Process, 2008. Available from http://menzies.us/pdf/08icsp.pdf.
37. J. Hihn, T. Menzies, K. Lum, T. Menzies, D. Baker, and O. Jalali. 2CEE, a Twenty First Century Effort Estimation Methodology. In ISPA'08: International Society of Parametric Analysis, 2008. Available from http://menzies.us/pdf/08ispa.pdf.
38. K. Gundy-Burlet, J. Schumann, T. Menzies, and T. Barrett. Parametric analysis of antares reentry guidance algorithms using advanced test generation and data analysis. In 9th International Symposium on Artifical Intelligence, Robotics and Automation in Space, 2008. Available from http://menzies.us/pdf/08antares.pdf.
39. T. Menzies, O. Elrawas, J. Hihn, M. Feathear, B. Boehm, and R. Madachy. The business case for automated software engineerng. In ASE '07: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, pages 303–312, New York, NY, USA, 2007. ACM.
40. Y. Jiang, B. Cukic, and T. Menzies. Fault prediction using early lifecycle data. In ISSRE'07, 2007. Available from http://menzies.us/pdf/07issre.pdf.
41. J.H. Andrews, F.C.H. Li, and T. Menzies. Nighthawk: A two-level genetic-random unit test data generator. In IEEE ASE'07, 2007. Available from http://menzies.us/pdf/ 07ase-nighthawk.pdf.
42. T. Menzies and Y. Hu. Agents in a wild world. In C. Rouff, M. Hinchey, J. Rash, W. Truszkowski, and D. Gordon-Spears, editors, Agent Technology from a Formal Perspective. Springer, 2006. Available from http://menzies.us/pdf/01agents.pdf.
43. K. Lum, J. Hihn, and T. Menzies. Sudies in software cost model behavior: Do we really understand cost model performance? In ISPA Conference Proceedings, 2006. Available from http:// menzies.us/pdf/06ispa.pdf.
44. J. Gao, M. Heimdahl, D. Owen, and T. Menzies. On the distribution of property violations in formal models: An initial study. In COMPSAC '06, 2006. Available from http://menzies. us/pdf/06compsac.pdf.
45. M.S. Fisher and T. Menzies. Learning ivv strategies. In HICSS'06, 2006. Available from http: //menzies.us/pdf/06hicss.pdf.
46. T. Menzies and J. Richardson. Xomo: Understanding development options for autonomy. In COCOMO forum, 2005, 2005. Available from http://menzies.us/pdf/05xomo_cocomo_ forum.pdf. For more details, see also the longer technical report http://menzies.us/ pdf/05xomo101.pdf.

47. T. Menzies, D. Port, Z. Chen, J. Hihn, and S. Stukes. Validation methods for calibrating software effort models. In Proceedings, ICSE, 2005. Available from http://menzies.us/pdf/ 04coconut.pdf.
48. T. Menzies, D. Port, Z. Chen, J. Hihn, and S. Stukes. Specialization and extrapolation of induced domain models: Case studies in software effort estimation. In IEEE ASE, 2005, 2005. Available from http://menzies.us/pdf/05learncost.pdf.
49. David Owen, Tim Menzies, Mats Heimdahl, and Jimin Gao. On the advantages of approximate vs. complete verification: Bigger models, faster, less memory, usually accurate. In IEEE NASE SEW 2003, 2003. Available from http://menzies.us/pdf/03lurchc.pdf.
50. D. Owen and T. Menzies. Lurch: a lightweight alternative to model checking. In SEKE '03, 2003. Available from http://menzies.us/pdf/03lurch.pdf.
51. Tim Menzies and Justin S. Di Stefano. How good is your blind spot sampling policy? In 2004 IEEE Conference on High Assurance Software Engineering, 2003. Available from http:// menzies.us/pdf/03blind.pdf.
52. Tim Menzies, Robyn Lutz, and Carmen Mikulski. Better analysis of defect data at NASA. In SEKE03, 2003. Available from http://menzies.us/pdf/03superodc.pdf.
53. T. Menzies, J.S. Di Stefano, and M. Chapman. Learning early lifecycle IVV quality indicators. In IEEE Metrics '03, 2003. Available from http://menzies.us/pdf/03early.pdf.
54. Yan Liu, Srikanth Gururajan, Bojan Cukic, Tim Menzies, and Marcello Napolitano. Validating an online adaptive system using svdd. In IEEE Tools with AI, 2003. Available from http: //menzies.us/pdf/03svdd.pdf.
55. D. Geletko and T. Menzies. Model-based software testing via treatment learning. In IEEE NASE SEW 2003, 2003. Available from http://menzies.us/pdf/03radar.pdf.
56. M.S. Feather, T. Menzies, and J.R. Connelly. Relating practitioner needs to research activities, September 2003. Available from http://menzies.us/pdf/03ieeere.pdf.
57. M.S. Feather, T. Menzies, and J.R. Connelly. Matching software practitioner needs to researcher activities. In Proceedings of the 2003 Asia-Pacific Software Engineering Conference (APSEC 2003); Chiangmai, Thailand. December 2003. Available from http://menzies.us/pdf/ 03iemc.pdf.
58. M.S. Feather, T. Menzies, and J.R. Connelly. Identifying fruitful connections between and among researchers and practitioners. In Proceedings of the 2003 IEEE International Engineering Management Conference (IEMC-2003) on Managing Technologically Driven Organizations; Albany, NY,, pages 451–455. November 2003. Available from http://menzies.us/pdf/03iemc.pdf.
59. S. L. Cornford, M. S. Feather, J.R. Dunphy, J. Salcedo, and T. Menzies. Optimizing spacecraft design optimization engine development: Progress and plans. In Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, 2003. Available from http://menzies.us/pdf/03aero. pdf.
60. E. Chiang and T. Menzies. Position paper: Summary of simulations for very early lifecycle quality evaluations. In Prosim '03, 2003. Available from http://menzies.us/pdf/03prosim. pdf.
61. J.S. Di Stefano and T. Menzies. Machine learning for software engineering: Case studies in software reuse. In Proceedings, IEEE Tools with AI, 2002, 2002. Available from http:// menzies.us/pdf/02reusetai.pdf.
62. D. Owen, T. Menzies, and B. Cukic. What makes finite-state models more (or less) testable? In IEEE Conference on Automated Software Engineering (ASE '02), 2002. Available from http: //menzies.us/pdf/02moretest.pdf.
63. Tim Menzies, David Raffo, Siri on Setamanit, Ying Hu, and Sina Tootoonian. Model-based tests of truisms. In Proceedings of IEEE ASE 2002, 2002. Available from http://menzies.us/ pdf/02truisms.pdf.
64. T. Menzies, D. Owen, and B. Cukic. Saturation effects in testing of formal models. In ISSRE 2002, 2002. Available from http://menzies.us/pdf/02sat.pdf.
65. T. Menzies and L. Mason. Some prolog macros for rule-based programming: Why? how? In Third ACM SIGPLAN Workshop on Rule-Based Programming (RULE02) Pittsburgh, PA, October 5, 2002. Available from http://menzies.us/pdf/03datasniffing.pdf.
66. Y. Liu, T. Menzies, and B. Cukic. Data sniffing - monitoring of machine learning for online adaptive systems. In IEEE Tools with AI, 2002. Available from http://menzies.us/pdf/ 03datasniffing.pdf.
67. M.S. Feather and T. Menzies. Converging on the optimal attainment of requirements. In IEEE Joint Conference On Requirements Engineering ICRE'02 and RE'02, 9-13th September, University of Essen, Germany, 2002. Available from http://menzies.us/pdf/02re02.pdf.
68. T. Menzies, J. Powell, and M. E. Houle. Fast formal analysis of requirements via 'topoi diagrams'. In ICSE 2001, 2001. Available from http://menzies.us/pdf/00fastre.pdf.
69. T. Menzies and J.D. Kiper. Better reasoning about software engineering activities. In ASE-2001, 2001. Available from http://menzies.us/pdf/01ase.pdf.
70. Tim Menzies, Bojan Cukic, Harhsinder Singh, and John Powell. Testing nondeterminate systems. In ISSRE 2000, 2000. Available from http://menzies.us/pdf/00issre.pdf.
71. T. Menzies and E. Sinsel. Practical large scale what-if queries: Case studies with software risk assessment. In Proceedings ASE 2000, 2000. Available from http://menzies.us/pdf/ 00ase.pdf.

72. T.J. Menzies, S. Easterbrook, Bashar Nuseibeh, and Sam Waugh. An empirical investigation of multiple viewpoint reasoning in requirements engineering. In RE '99, 1999. Available from http://menzies.us/pdf/99re.pdf.
73. T. Menzies and C.C. Michael. Fewer slices of pie: Optimising mutation testing via abduction. In SEKE '99, June 17-19, Kaiserslautern, Germany., 1999. Available from http://menzies. us/pdf/99seke.pdf.
74. T. Menzies and B. Cukic. On the sufficiency of limited testing for knowledge based systems. In The Eleventh IEEE International Conference on Tools with Artificial Intelligence. November 9-11, 1999. Chicago IL USA., 1999.
75. T.J. Menzies and S. Waugh. On the practicality of viewpoint-based requirements engineering. In Proceedings, Pacific Rim Conference on Artificial Intelligence, Singapore. Springer-Verlag, 1998. Available from http://menzies.us/pdf/98pracai.pdf.
76. M. Postema, T.J. Menzies, and X. Wu. A decision support tool for tuning parameters in a machine leraning algorithm. In The Joint Pacific Asia Conference on Expert Systems/Singapore International Conference on Intelligent Systems. (PACES/SPICIS '97), 1997. Available from http://menzies.us/pdf/97pakdd.pdf.
77. M.Posterma, X. Wu, and T.J. Menzies. A tuning aid for discretization in rule induction. In First Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD97), 1997. Available from http://menzies.us/pdf/97paces.pdf.
78. S Ramakrishnan, T. Menzies, M. Hasslinger, P. Bok, H. McCarthy, B. Devakadadcham, and D. Moulder. On building an effective measurement system for oo software process, product and resource tracking. In Tools Pacific, 1996, 1996.
79. S. Ramakrishnan, T. Menzies, M. Hasslinger, P. Bok, H. Mccarthy, B. Devakadadcham, and D. Moulder. On building an effective measurement system for oo software process. In Proceedings of Tools-Pacific, Melbourne. Prentice-Hall, 1996. Available from http://menzies. us/pdf/96process.pdf.
80. S Ramakrishnan and T. Menzies. An ongoing experiment in o-o software process and product measurements. In Proceedings SEEP'96, New Zealand, 1996.
81. T.J. Menzies. Visual programming, knowledge engineering, and visual programming. In Proceedings of the Eighth International Conference on Software Engineering and Knowledge Engineering. Knowledge Systems Institute, Skokie, Illinois, USA, 1996. Available from http: //menzies.us/pdf/96seke.pdf.
82. T.J. Menzies. On the practicality of abductive validation. In ECAI '96, 1996. Available from http:/menzies.us/pdf/96ok.pdf.
83. T. Menzies and S. Ramakrishnan. Comparing and generalising models for metrics repositories. In Tools Pacific, Melbourne, 1996. Available from http://menzies.us/pdf/96metrics. pdf.
84. M. Connell and T.J. Menzies. Quality metrics: Test coverage analysis for smalltalk. In Tools Pacific, 1996, Melbourne, 1996. Available from http://menzies.us/pdf/96conel.pdf.
85. R.F. Cohen and T. J. Menzies. Providing Software Engineering Students with an Experience in "Big-Computing". In Software Education Conference (SRIG-ET'94), pages 71–76, 1995.
86. T.J. Menzies and P. Haynes. The Methodologies of Methodologies; or, Evaluating Current Methodologies: Why and How. In Tools Pacific '94, pages 83–92. Prentice-Hall, 1994. Available from http://menzies.us/pdf/tools94.pdf.
87. P. Haynes and T.J. Menzies. The Effects of Class Coupling on Class Size in Smalltalk Systems. In Tools '94, pages 121–129. Prentice Hall, 1994.
88. T.J Menzies and R Spurret. How to Edit ¨t¨ or a Black-box Constraint Based Framework for User ı; Interaction with Arbitrary Structures. In Tools Pacific 12, pages 213–224. Prentice Hall, 1993. Available from http://menzies.us/pdf/tools93.pdf.
89. P. Haynes and T.J. Menzies. C++ is Better than Smalltalk? In Tools Pacific 1993, pages 75–82, 1993.
90. T.J. Menzies, J. Edwards, and K. Ng. The Mysterious Case of the Missing Re-usable Class Libraries. In Tools Pacific 1992, pages 421–428. Prentice Hall, 1992. Available from http: //menzies.us/pdf/tools92.pdf.
91. T.J. Menzies, J. Black, J. Fleming, and M. Dean. An expert system for raising pigs. In The first Conference on Practical Applications of Prolog, 1992. Available from http://menzies.us/ pdf/ukapril92.pdf.
92. T.J. Menzies. ISA Object PARTOF Knowledge Representation (part two)? In B. Meyer, editor, Tools Pacific 4, 1991. Available from http://menzies.us/pdf/tools91.pdf.
93. T.J. Menzies. Beyond the mvc triad: Quality assurance via interactive specification editors. In Tools 3: Proceedings of the third International Technology of Object-Oriented Languages and; Systems conference. Prentice-Hall, 1991.
94. Parametric analysis of a hover test vehicle using advanced test generation and data analysis.
95. T. Menzies and H. Singh. How AI can help SE; or: Randomized search not considered harmful. In AI'2001: the Fourteenth Canadian Conference on Artificial Intelligence, June 7-9, Ottawa, Canada, 2001. Available from http://menzies.us/pdf/00funnel.pdf.
96. S. Waugh, J. Blogs, and T. Menzies. The temporal qualitative compartmental modeling language. In Proceedings of the Australain AI '98 conference, 1998. Available from http://menzies. us/pdf/97links.pdf.
97. T.J. Menzies and S. Waugh. Lower limits on the size of test data sets. In Proceedings of the Australian AI '98 conference. World-Scientific, 1998. Available from http://menzies.us/ pdf/98ozai.pdf.

98. S. Waugh, T.J. Menzies, and S. Goss. Evaluating a qualitative reasoner. In Abdul Sattar, editor, Advanced Topics in Artificial Intelligence: 10th Australian Joint Conference on AI. Springer-Verlag, 1997. http://www.cse.unsw.edu.au/˜timm/pub/docs.
99. T.J. Menzies. Situated Semantics is a Side-Effect of the Computational Complexity of Abduction. In Australian Cognitive Science Society, 3rd Conference, 1995. Available from http: //menzies.us/pdf/cogsci95.pdf.
100. T.J. Menzies. Limits to Knowledge Level-B Modeling (and KADS). In Proceedings of AI '95, Australia. World-Scientific, 1995. Available from http://menzies.us/pdf/95akads. pdf.
101. T.J. Menzies and P. Compton. A Precise Semantics for Vague Diagrams. In C. Zhang, J. Debenham, and D. Lukose, editors, Proceedings of Australian AI'94, pages 149–156. World Scientific, 1994. Available from http://menzies.us/pdf/ai94.pdf.
102. T.J. Menzies. Maintaining procedural knowledge: Ripple-down-functions. In Proceedings of AI '92, Australia, 1992. Available from http://menzies.us/pdf/ai92.pdf.
103. A.J. Mahidadia, P. Compton, T.J. Menzies, C. Sammut, and G.A. Smythe. Inventing causal qualitative models: A tool for experimental research. In AI '92, Horbart, Australia. World-Scientific, 1992.
104. T.J. Menzies. Isa object part-of knowledge representation? In Proceedings AI '90, 1990.
105. T.J. Menzies, M. Dean, J. Black, and J. Fleming. Combining heuristics with simulation models: An expert system for the optimal management of pig. In AI '88, 1988. Adelaide, Australia.
106. T.J. Menzies and C. Worral. Worlds in prolog. In Proceedings of AI '87, 1987.
107. T.J. Menzies and B.R. Markey. A micro-computer, rule-based prolog expert-system for process control in a petrochemical plant. In Proceedings of the Third Australian Conference on Expert Systems, May 13-15, 1987.

High Impact, Non-Refereed Publications

1. **Tim Menzies, Cross-Project Data for Software Engineering, IEEE Computer December, 2015, p6**
2. **Bird, Christian, Timothy Menzies, and Thomas Zimmermann. "Past, Present, and Future of Analyzing Software Data." Elsevier Inc.. 2015 in The Art and Science of Analyzing Software Data C. Bird, T. Menzies, T. Zimmermann, Morgan Kaufmann, 2016**
3. **Tim Menzies, Corina Pasareanu, Guest editorial: special multi-issue on selected topics in Automated Software Engineering. Automated Software Engineering journal, 22(3) 289-290, 2015**
4. **R. Harrison, T. Menzies, Guest editorial: special issue on realizing AI synergies in software engineering, Automated Software Engineering, 22(1), 2015**
5. Tim Menzies: Beyond Data Mining. IEEE Software 30(3): 92 (2013)
6. Tim Menzies, Thomas Zimmermann: Software Analytics: So What? IEEE Software 30(4): 31-37 (2013)
7. Tim Menzies, Thomas Zimmermann: The Many Faces of Software Analytics. IEEE Software 30(5): 28-29 (2013)
8. Tim Menzies: Guest editorial for the Special Section on BEST PAPERS from the 2011 conference on Predictive Models in Software Engineering (PROMISE). Information & Software Technology 55(8): 1477-1478 (2013)
9. Tim Menzies, Martin Shepperd: Special issue on repeatable results in software engineering prediction. Empirical Software Engineering 17(1-2): 1-17 (2012)
10. T. Menzies. 21st century AI: proud, not smug. IEEE Intelligent Systems, 2003. Available from http://menzies.us/pdf/03aipride.pdf.
11. T. Menzies and F. van Harmelen. Editorial: Evaluating knowledge engineering techniques. International Journal of Human-Computer Studies, special issue on evaluation of Knowledge Engineering Techniques, 51(4):717–727, October 1999. Available from http://menzies.us/pdf/ 99ekeed.pdf.
12. T. Menzies. Knowledge maintenance heresies: Meta-knowledge complicates km. In 11th Annual International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, June 17 - 19, 1999, 1999. Available from http://menzies.us/pdf/99sekekm. pdf.
13. T. Menzies. Desert island column. Automated Software Engineering, 6(3):315–320, 1999. Available from http://menzies.us/desert.html.

Other Publications (workshops, etc)

1. **Data Mining Methods and Cost Estimation Models: Why is it So Hard to Infuse New Ideas? Jairus Hihn, Tim Menzies, 2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)**
2. Learning the task management space of an aircraft approach model, AAAI 2014 Spring Symposium, Joseph Krall, Tim Menzies, Misty Davis.
3. Replication in Empirical Software Engineering Research (RESER). On parameter tuning in search based software engineering: A replicated empirical study, AS Sayyad, K Goseva-Popstojanova, T Menzies, H Ammar, 2013

4. Rachel Harrison, Daniela Carneiro da Cruz, Pedro Rangel Henriques, Maria João Varanda Pereira, Shih-Hsi Liu, Tim Menzies, Marjan Mernik, Daniel Rodríguez:Report from the first international workshop on realizing artificial intelligence synergies in software engineering (RAISE 2012). ACM SIGSOFT Software Engineering Notes 37(5): 34-35 (2012)
5. Menzies, Tim and Bird, Christian and Zimmermann, Thomas and Schulte, Wolfram and Kocaganeli, Ekrem. The inductive software engineering manifesto: principles for industrial data mining by Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering 19--26 2011 .
6. B. Turhan, A. Bener, and T. Menzies. Nearest neighbor sampling for cross company defect predictors. In Proceedings, DEFECTS 2008, 2008. hW.
7. T. Menzies, B. Turhan, A. Bener, G. Gay, B. Cukic, and Y. Jiang. Implications of ceiling effects in defect predictors. In Proceedings of PROMISE 2008 Workshop (ICSE), 2008. Available from http://menzies.us/pdf/08ceiling.pdf.
8. Y. Jiang, B. Cukic, T. Menzies, and N. Bartlow. Comparing design and code metrics for software quality prediction. In Proceedings of the PROMISE 2008 Workshop (ICSE), 2008. Available from http://menzies.us/pdf/08compare.pdf.
9. Y. Jiang, B. Cukic, and T. Menzies. Does transformation help? In Defects 2008, 2008. Available from http://menzies.us/pdf/08transform.pdf.
10. T. Menzies, O. Elrawas, D. Baker, J. Hihn, and K. Lum. On the value of stochastic abduction (if you fix everything, you lose fixes for everything else). In International Workshop on Living with Uncertainty (an ASE'07 co-located event), 2007. Available from http://menzies.us/pdf/ 07fix.pdf.
11. T. Menzies, D. Allen, and A. Orrego. Bayesian anomaly detection (bad v1.0). In Proceedings of the Machine Learning Algorithms for Surveillance and Event Detection Workshop, ICML'06, 2006. Available from http://menzies.us/pdf/06bad.pdf.
12. T. Menies, K. Lum, and J. Hihn. The deviance problem in effort estimation. In PROMISE, 2006, 2006. Available from http://menzies.us/06deviations.pdf.
13. M. . Feather, S.. Cornford, J. Kiper, and T. Menzies. Experiences using visualization techniques to present requirements, risks to them, and options for risk mitigation. In First International Workshop on Requirements Engineering Visualization, 2006. Available from http: //menzies.us/pdf/06rev.pdf.
14. Tim Menzies, Zhihao Chen, Dan Port, and Jairus Hihn. Simple software cost estimation: Safe or unsafe? In Proceedings, PROMISE workshop, ICSE 2005, 2005. Available from http: //menzies.us/pdf/05safewhen.pdf.
15. Zhihoa Chen, Tim Menzies, and Dan Port. Feature subset selection can improve software cost estimation. In PROMISE'05, 2005. Available from http://menzies.us/pdf/05/ fsscocomo.pdf.
16. T. Menzies, Justin S. Di Stefano, Chris Cunanan, and Robert (Mike) Chapman. Mining repositories to assist in project planning and resource allocation. In International Workshop on Mining Software Repositories, 2004. Available from http://menzies.us/pdf/04msrdefects.pdf.
17. T. Menzies, S. Setamanit, and D. Raffo. Data mining from process models. In PROSIM 2004, 2004. Available from http://menzies.us/pdf/04dmpm.pdf.
18. T. Menzies, J. DiStefano, A. Orrego, and R. Chapman. Assessing predictors of software defects. In Proceedings, workshop on Predictive Software Models, Chicago, 2004. Available from http: //menzies.us/pdf/04psm.pdf.
19. A. Dekhtyar, J. Huffman Hayes, and T. Menzies. Text is software too. In International Workshop on Mining Software Repositories (submitted), 2004. Available from http://menzies.us/ pdf/04msrtext.pdf.
20. T. Burkleaux, T. Menzies, and D. Owen. Lean = (lurch+tar3) = reusable modeling tools. In Proceedings of WITSE 2005, 2004. Available from http://menzies.us/pdf/04lean. pdf.
21. T. Menzies, J. Kiper, and M. Feather. Improved software engineering decision support through automatic argument reduction tools. In SEDECS'2003: the 2nd International Workshop on Software Engineering Decision Support (part of SEKE2003), June 2003. Available from http: //menzies.us/pdf/03star1.pdf.
22. Tim Menzies, Justin S. DiStefeno, Mike Chapman, and Kenneth Mcgill. Metrics that matter. In 27th NASA SEL workshop on Software Engineering, 2002. Available from http://menzies. us/pdf/02metrics.pdf.
23. T. Menzies, A. Pearce, C. Heinze, and S. Goss. What is an agent and why should i care? In Formal Aspects of Agent-Based Systems, 2002. Available from http://menzies.us/pdf/ 02agentis.pdf.
24. T. Menzies, D. Owen, and B. Cukic. You seem friendly, but can i trust you? In Formal Aspects of Agent-Based Systems, 2002. Available from http://menzies.us/pdf/02trust.pdf.
25. D. Owen and T. Menzies. Random search of and-or graphs representing finite-state models. In Proceedings of the First International Workshop on Model-based Requirements Engineering, 2001. Available from http://menzies.us/pdf/01randandor.pdf.
26. T. Menzies and H. Singh. Many maybes mean (mostly) the same thing. In 2nd International Workshop on Soft Computing applied to Software Engineering (Netherlands), February, 2001. Available from http://menzies.us/pdf/00maybe.pdf.

27. T. Menzies and Y. Hu. Reusing models for requirements engineering. In First International Workshop on Model-based Requirements Engineering, 2001. Available from http://menzies. us/pdf/01reusere.pdf.
28. T. Menzies and Y. Hu. Constraining discussions in requirements engineering. In First International Workshop on Model-based Requirements Engineering, 2001. Available from http: //menzies.us/pdf/01lesstalk.pdf.
29. T. Menzies and B. Cukic. Average case coverage for validation of ai systems. In AAAI Stanford Spring Symposium on Model-based Validation of AI Systems, 2001. Available from http:// menzies.us/pdf/01validint.pdf.
30. T.J. Menzies. The complexity of trmcs-like spiral specification. In Proceedings of 10th International Workshop on Software Specification and Design (IWSSD-10), 2000. Available from http://menzies.us/pdf/00iwssd.pdf.
31. Tim Menzies, Bojan Cukic, and Harhsinder Singh. Agents talking faster, April 2000. NASA Goddard Workshop on Formal Aspects of Agent-Oriented Systems. Available from http:// menzies.us/pdf/00godd.pdf.
32. T. Menzies, E. Sinsel, and T. Kurtz. Learning to reduce risks with cocomo-ii. In Workshop on Intelligent Software Engineering, an ICSE workshop, and NASA/WVU Software Research Lab, Fairmont, WV, Tech report # NASA-IVV-99-027, 2000. Available from http://menzies.us/ pdf/00wise.pdf.
33. T. Menzies and B. Cukic. Maintaining maintainability = recognizing reachability. In International Workshop on Empirical Studies of Software Maintenance (WESS 2000), October 14, San Jose CA, 2000. Available from http://menzies.us/pdf/00wess.pdf.
34. T. Menzies, B. Cukic, and E. Coiera. Smaller, faster dialogues via conversational probing. In AAAI'99 workshop on Conflicts and Identifying Opportunities., 1999. Available from http: //menzies.us/pdf/99aaaic.pdf.
35. T. Menzies and B. Cukic. Intelligent testing can be very lazy. In Proceedings, AAAI '99 workshop on Intelligent Software Engineering, Orlando, Florida, July 1999. Available from http:// menzies.us/pdf/99waaai.pdf.
36. T. Menzies. hQkb- the high quality knowledge base initiative (sisyphus v: Learning design assessment knowledge). In KAW'99: the 12th Workshop on Knowledge Acquisition, Modeling and Management, Voyager Inn, Banff, Alberta, Canada Oct 16-22, 1999, 1999. Available from http://menzies.us/pdf/99hqkb.pdf.
37. D. Richards and T.J. Menzies. Extending the sisyphus iii experiment from a knowledge engineering task to a requirements engineering task. In Banff Workshop on Knowledge Acquisition, 1998. Available from http://menzies.us/pdf/98kawre.pdf.
38. T.J. Menzies and S. Waugh. More results on the practical lower limits of test set size. In Proceedings Pacific Knowledge Acquisition Workshop, Singapore, November, 1998, 1998. Available from http://menzies.us/pdf/98pkaw.pdf.
39. T.J. Menzies, R.F. Cohen, and S. Waugh. Evaluating conceptual qualitative modeling languages. In Banff KAW '98 workshop., 1998. Available from http://menzies.us/pdf/97modlan. pdf.
40. T.J. Menzies. Evaluation issues with critical success metrics. In Banff KA '98 workshop., 1998. Available from http://menzies.us/pdf/97langevl.pdf.
41. T.J. Menzies. Evaluation issues for problem visual programming languages, 1998. Banff KA workshop, 1998. Available from http://menzies.us/pdf/97evalvp.pdf.
42. T.J. Menzies. Evaluation issues for problem solving methods. In Banff Knowledge Acquisition workshop, 1998, 1998. Available from http://menzies.us/pdf/97eval.pdf.
43. T. Menzies. Applications of abduction: A unified framework for software and knowledge engineering. Asian-Pacific Workshop on Intelligent Software Engineering, 1998. Available from http://menzies.us/pdf/98apwise.pdf.
44. D. Richards and T.J. Menzies. Extending knowledge engineering to requirements engineering from multiple perspectives. In T.J. Menzies, D. Richards, and P. Compton, editors, Third Australian Knowledge Acquisition Workshop, Perth, 1997. Available from http://menzies.us/pdf/ 97akawre.pdf.
45. T.J. Menzies and A. Mahidadia. Ripple-down rationality: A framework for maintaining psms. In Workshop on Problem-Solving Methods for Knowledge-based Systems, IJCAI '97, August 23., 1997. Available from http://menzies.us/pdf/97rdra.pdf.
46. T.J. Menzies and R.E. Cohen. A graph-theoretic optimisation of temporal abductive validation. In European Symposium on the Validation and Verification of Knowledge Based Systems, Leuven, Belgium, 1997. Available from http://menzies.us/pdf/97eurvav.pdf.
47. T.J. Menzies and S. Goss. Vague models and their implications for the kbs design cycle. In Proceedings PKAW '96: Pacific Knowledge Acquisition Workshop and Monash University Department of Software Development Technical Report TR96-15, 1996. Available from http: //menzies.us/pdf/96abmod.pdf.
48. T.J. Menzies. Assessing responses to situated congition. In Proceedings of the 10th Knowledge Acquisition Workshop for Knowledge-Based Systems, Banff,Canada, 1996. Available from http: //menzies.us/pdf/96sitcog.pdf.
49. Tim Menzies. Expert systems inference = modeling conflicts. In Proceedings of the ECAI '96 workshop on Modelling Conflicts in AI, 1996. Available from http://menzies.us/pdf/ 96ecaimc.pdf.
50. T. Menzies. Generalised test = generalised inference. In Proceedings of the ECAI '96 workshop on Validation, Verification, and Refinement of KBS, 1996. Available from http://menzies. us/pdf/96ecaivv.pdf.

51. T.J. Menzies and S. Goss. Applications of abduction #3: “black-box” to “gray-box” model. In AI in Defence Workshop, Australian AI’95, also Technical Report TR95-31, Department of Software Development, Monash University, 1995. Available from http://menzies.us/pdf/ 95gray.pdf.
52. T.J. Menzies and P. Compton. The (extensive) implications of evaluation on the development of knowledge-based systems.
53. In Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge Based Systems,, 1995. Available from http://menzies.us/pdf/ banff95.pdf.
54. P. Haynes, T. Menzies, and G. Phipps. Using the size of classes and methods as the basis for early effort prediction; empirical observations, initial application; a practitioners experience report. In OOPSLA Workshop on OO Process and Metrics for Effort Estimation, 1995.
55. T.J. Menzies and W. Gambetta. Exhaustive Abduction: A Practical Model Validation Tool. In ECAI ’94 Workshop on Validation of Knowledge-Based Systems, 1994. Available from http: //menzies.us/pdf/ecai94.pdf.
56. T.J. Menzies and P. Compton. Knowledge acquisition for performance systems; or: When can ”tests” replace ”tasks”? In Proceedings of the 8th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, 1994. http://menzies.us/pdf/ banff94.pdf.
57. T.J. Menzies. The complexity of model review. In DX-93: The International Workshop on Principles on Model-Based Diagnosis, 1993.
58. T.J. Menzies, P. Compton, and A. Mahidadia. Communicating research models of human physiology using qualitative compartmental modeling. In Communicating Scientific and Technical Knowledge, an AAAI ’92 workshop, 1992.
59. T.J. Menzies, P. Compton, B. Feldman, and T. Toft. Qualitative compartmental modeling. In Proceedings of the AAAI Symposium on Diagrammatic Reasoning Stanford University, March 2527, 1992.
60. T.J. Menzies and P. Compton. Causal explanations as a tool for refining qualitative models. In ECAI ’92 Workshop on Improving the Use of Knowledge-Based Systems with Explanations, Vienna, 1992.
61. T. Menzies, A. Mahidadia, and P. Compton. Using causality as a generic knowledge representation, or why and how centralised knowledge servers can use causality. In Proceedings of the 7th AAAISponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, 1992.
62. T. Menzies, A. Mahidadia, and P. Compton. Using Causality as a Generic Knowledge Representation, or Why and How Centralised Knowledge Servers Can Use Causality. In Proceedings of the 7th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop Banff, Canada, October 11-16, 1992.
63. T.J. Menzies. Concerning the user of procedural construct as a knowledge acquisition technique. In IJCAI ’91 Knowledge Acquisition Workshop, 1991.
64. P. Compton, G. Edwards, B. Kang, L. Lazarus, R. Malor, T. Menzies, P. Preston, A. Srinivasan, and C. Sammut. Ripple down rules: possibilities and limitations. In 6th Banff AAAI Knowledge Acquisition for Knowledge Based Systems, 1991
65. T. Menzies. Applictions of computational intelligence to quantitative software engineering, 2001. Available from http://menzies.us/pdf/01quase.pdf.
66. T.J. Menzies. Qualitative causal diagrams for requirements engineering. In The Second Australian Workshop on Requirements Engineering (AWRE’97), 1997. Available from http://menzies. us/pdf/97awre.pdf.
67. T. J. Menzies. Applications of abduction: Intelligent decision support systems. In Proceedings of the Melbourne Workshop on Intelligent Decision Support. Department of Information Systems, Monash University, Melbourne, 1996. Available from http://menzies.us/pdf/95idss.pdf.
68. T.J. Menzies. Applications of abduction #1: Intelligent decision support systems. In Proceedings of the Melbourne Workshop on Intelligent Decision Support Department of Information Systems Monash University, Caulfield Campus, Melbourne Monday, March 20, 1995, 1995. Available from http://menzies.us/pdf/95idss.pdf.

B. RESEARCH FUNDING

Total $8,074,703 (includes $210,000 in gifts from industry)

New funding for this year: $759,622

Proposal submitted (and declined):

- **NSF: Changing Software to Reduce Defects: $249,594**

Proposals submitted (under review):

- **NSF: Holistic Scalable Autotuning for Software Engineer Data Analytics, with Xipeng Shen, $1,200,000**
- **NSF: Attack Surfaces to Find Reachable Code Vulnerabilities, with Laruie William $ 499,948**

*Sponsored (new amounts for this year in **ORANGE**. On-going from last year in **RED**)*

start	finish	Funding body	Name	gift?	grant: sole PI	co-PI (expended by TM:)	Resaerch expenditure	Grants-Co-PI (total)	Research expenditure	current ?	new for 2015-2016
					A	B	C=A + B	D	E=C+D		
2016	2016	IBM	Automated software engineering	y	$40,000	$0	40,000	$0	$40,000	yes	yes
2016	2018	NSF	Reu: Science of Software			$10,000	10,000	$345,365	$355,365	yes	yes
2016	2016	Lexis Nexis	Optimization of ML for Big Data	y	$50,000	$0	50,000	$0	$50,000	yes	yes
2016	2016	SEI	Optimization business process		$75,000	$0	75,000	$0	$75,000	yes	yes
2015	2017	NCDSA	Share Care Beware			$60,000	60,000	$59,257	$119,257	yes	yes
2015	2018	Lexis Nexis	Validation lab	y	$120,000		120,000		$120,000	yes	yes
2015	2015	JPL	Effort Estimation (year2)		$30,000		30,000		$30,000		
2014	2015	Lexis Nexis	Scripting for Big data		$50,000		50,000		$50,000		
2013	2017	NSF	Transfer Learning in SE			622,030	622,030	529,773	$1,151,803	yes	
2013	2014	NASA (JPL)	Effort estimation		47,000		47,000		$47,000		
2012	2016	USDA	Early Childhood Obesity Program		$133,526		$133,526		$133,526		
2012	2013	NSF	New directions in AI and SE		$14,700		$14,700		$14,700		
2010	2012	Dod STTRv)	Active Learning		$230,514		$230,514		$230,514		
2010	2014	NSF (CISE)	Better Understanding of SE data			$249,500	$249,500	$499,000	$748,500		
2010	2012	Qatar Resarch	Int Center of Excellence in SE			$98,125	$98,125	$196,250	$294,375		
2010	2011	CITRE	Border Crossing		$70,000		$70,000		$70,000		
2010	2011	National Forensics	Overcoming Brittleness		$35,721		$35,721		$35,721		
2009	2010	National Archives	STEP Research			$209,000	$209,000	$418,000	$627,000		
2008	2009	National archives	STEP research			$143,500	$143,500	$574,000	$717,500		
2008	2011	NSF (CISE)	Automatic Quality Assessment			$180,000	$180,000	$360,000	$540,000		
2008	2009	National Forensics	Conclusion stability		$80,000		$80,000		$80,000		
2008	2008	NASA	Understanding Anomalies.		$58,000		$58,000		$58,000		
2008	2008	NASA	Crystal Ball.		$55,000		$55,000		$55,000		
2008	2008	NASA	Advanced UML modeling.		$50,000		$50,000		$50,000		
2007	2008	NASA	Applied Technology Lab		$95,551		$95,551		$95,551		
2007	2008	Dod STTRv	Next generation metrics: phase 1		$40,715		$40,715		$40,715		
2007	2007	NASA	WVU Liaison		$39,707		$39,707		$39,707		
2007	2008	Industrial	Analysis metrics (Galaxy Global)		$25,000		$25,000		$25,000		
2007	2008	National archives	STEP research		$15,482		$15,482		$15,482		
2006	2007	NASA	Learning software process model		$113,255		$113,255		$113,255		
2006	2007	NASA	Improving IV&V Techniques		$107,990		$107,990		$107,990		
2006	2006	NASA	co-op agreement supplemental funds			$14,916	$14,916	$59,665	$74,581		
2006	2006	NASA	co-op funds for Eisland Hall Lab		$30,000		$30,000		$30,000		
			sum (2006 to 2015)		$1,607,161	$1,587,071	$3,194,232	$3,041,310	$6,235,542		

2005	2005	NASA	How to Argue Less:		$260,000		$260,000		$260,000
2005	2005	NASA	Spectrum of Model Checking Methods		$160,000		$160,000		$160,000
2005	2005	NASA	Risk/Cost models for Autonomy		$160,000		$160,000		$160,000
2005	2005	NASA	How much will it cost?		$122,161		$122,161		$122,161
2005	2005	NASA SBIT	Intelligent Vehicle Health Management:		$65,000		$65,000		$65,000
2004	2004	NASA	Spectrum of Model Checking Methods		$160,000		$160,000		$160,000
2004	2005	NASA	A next-generation testable language		$70,000		$70,000		$70,000
2004	2004	NASA	The research rover		$48,000		$48,000		$48,000
2003	2005	NASA	Understanding models better		$107,000		$107,000		$107,000
2003	2003	NASA	Model checking & procedural languages		$50,000		$50,000		$50,000
2003	2003	NASA	See more! Learn more! Tell more!		$47,000		$47,000		$47,000
2002	2003	NASA	A spectrum of IV&V techniques		$200,000		$200,000		$200,000
2002	2002	NASA	Better risk modelling		$27,000		$27,000		$27,000
2001	2001	NASA	Tree query languages		$27,000		$27,000		$27,000
2000	2000	Canada Res. Coun.	NSERC grant		$81,000		$81,000		$81,000
1998	1999	NASA	High Quality Knowledge Initiative		$110,000		$110,000		$110,000
1997	1998	Aust. Res. Coun	Abduction for software engineering		$10,000		$10,000		$10,000
1996	1998	UNSW	Vice-Chancellor's Research Fellowship		$135,000		$135,000		$135,000
			Total (1996 to 2005)		$1,839,161	$0	$1,839,161	$0	$1,839,161
			Total (ALL)		$3,446,322	$1,587,071	$5,033,393	$3,041,310	$8,074,703

External Funding						
2015-0916	Provide Support in Developing Cost estimating models for the NASA Software CER Development Task	Menzies, Timothy James	Computer Science	Jet Propulsion Laboratory (Prime - National Aeronautics & Space Administration (NASA))	$28,500	04/10/2015 through 01/31/2016
2015-0943	SHF:Medium:Collaborative:Transfer Learning in Software Engineering	Menzies, Timothy James	Computer Science	National Science Foundation (NSF)	$316,681	08/02/2014 through 06/30/2017
2016-0911	Enabling Evidence-Based Modernization	Menzies, Timothy James	Computer Science	Carnegie Mellon University (Prime - US Air Force (USAF))	$35,000	01/13/2016 through 09/30/2016
Total external funding: $380,181						

Internal Funding
Total internal funding: $0

Pending Proposals (including pre-proposals)					
2015-3234	Share, Care, Beware : Trusted Sharing Practices for Data Science	Menzies, Timothy James	Computer Science	North Carolina Data Science and Analytics Initiative (NCDSA)	$119,257
2016-0702	SHF:Medium:Holistic Scalable Autotuning for Software Engineer Data Analytics	Menzies, Timothy James Shen, Xipeng	Computer Science	National Science Foundation (NSF)	$1,200,000
2016-1357	TWC: Small: On the Practical Use of Attack Surfaces Find Reachable Code Vulnerabilities	Williams, Laurie A. Menzies, Timothy James	Computer Science	National Science Foundation (NSF)	$499,948
Total of pending proposals: $1,819,205					

Non-funded Projects					
2015-1051	CI-NEW: Next Generation Open Science Research for Software Engineering	Menzies, Timothy James Murphy-Hill, Emerson R	Computer Science	National Science Foundation (NSF)	$793,842
2015-1394	CPS: Synergy: Collaborative Research: Real Time Attack Monitoring and Control for Cyber Physical Security of Power Grid	Menzies, Timothy James	Computer Science	National Science Foundation (NSF)	$179,151
2015-1562	SHF: Small: Smarter Software Autotuning for SE Data Analytics	Menzies, Timothy James Shen, Xipeng	Computer Science	National Science Foundation (NSF)	$498,524
2015-1565	SHF:Small:Collaborative: Changing Software to Reduce Defects	Menzies, Timothy James	Computer Science	National Science Foundation (NSF)	$249,594
2016-0738	SHF:Medium:Collaborative Research: Changing Software to Reduce Defects	Menzies, Timothy James	Computer Science	National Science Foundation (NSF)	$740,607
2016-0934	Verifying Safety of NextGen Models: A Rational Approach	Menzies, Timothy James	Computer Science	National Aeronautics & Space Administration (NASA)	$0
Total of non-funded proposals: $2,461,718					

C. CENTERS AND MULTIDISCIPLINARY COLLABORATION

1. Curator, Promise Repository of SE Data: openscience.us/repo

   For artifacts other than code, this is now the largest open science resource in software engineering for long term storage of data used in publications on software analytics.

IV. EXTENSION AND ENGAGEMENT WITH CONSTITUENCIES OUTSIDE THE UNIVERSITY

- **Jet Propulsion Lab, effort estimation research. 2002 to present. Funded research 2007, 20014 to 2015**
- **IBM, data mining work with research triangle. Faculty award 2016.**
- **Lexis Nexis, text mining work, 2015, 2016.**
- Microsoft Research, research projects, February 2011 to 2012 to present
- NASA, Software Engineering Research Chair, 2001 to 2003
- NASA Effort Estimation research, 2004 to 2015
- Consultant, Object-oriented programming, 1988- 1995
- Consultant, Expert systems, 1985-1988

V. TECHNOLOGICAL AND MANAGERIAL INNOVATION

A. TECHNOLOGY TRANSFER

Workshop Organizer/Facilitator:

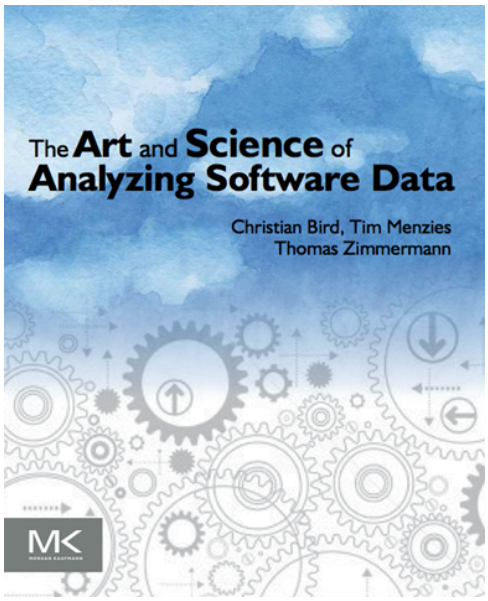
- **Big Data for SE, ICSE'16 workshop**
- Actionable Analytics, ASE'15 workshop
- Big Data for SE, ICSE'15 workshop
- RAISE'14 (Realizing AI Synergies with Software Engineering),, an ICSE 2014 workshop/
- Dagstuhl Seminar, Software Development Analytics, 2014 (co-organized with Laurie Williams and Tom Zimmermann).

Tutorial Presenter/Organizer:

- **ICSE 2016 Technical Briefing: How not to do it, Anti-Patterns in Data Analysis.**
- ICSE 2015 Technical Briefing: Art and Science of Analyzing Software Data (Quantitative Methods)
- ICSE 2014 Tutorial: Art and Science of Analyzing Software data
- ICSE 2013 Tutorial: Data Science for Software Engineering
- ICSE 2012 Tutorial: Understanding Machine Learning for Empirical Software Engineering
- June 2010: Data Mining summer school, Queens University, Kingston, Canada (http://goo.gl/oMcSX);
- Sept2010:LASERsummerschoolonempiricalsoftwareengineering,Elba,Italy(http://goo.gl/4lwDu).
- Feb 2010: Invited speaker, Microsoft, Empirical SE, version 2.0
- Sept 2008: Invited Speaker, Google, Defect Prediction

B. IMPACT

For my full career impact, see section B2 (below, p25).



B1. IMPACT, last 12 months

1. *What are the highlights of your achievements in the past year?*
    - Promoted the NCSU name as part of my work as co-General chair for the Raleigh conference on software maintenance and evolution
    - Research-wise: achieved landmark results in micro-sampling large spaces. Seem the world is NOT as complex as we think but we actually "walk" a low-dimensional manifold within which we can quickly make decisions (publications pending)
    - One new book, done "Art and Science of Analyzing Software Data". A technical summary of the state of the art in this active area written by hundreds of leading experts
    - Another book nearly done "Perspectives on Data Science for Software Engineering". Lessons learned from data science, express in terms accessible to a wide audience.
    - Continued the work on the PROMISE repository- the largest collection of SE data serving the international community-- all bannered under NCSU.

2. *In the last year, how have you had impact? Who have you impacted?*
    - Results from prior research work with the Jet Propulsion Laboratory now distributed internally within JPL as part of their cost estimation toolkit.
    - Working with IBM, research triangle
        - Advising them on how to organize their analytics (that work recognized with an IBM Faculty award)
    - Working with ABB, Raleigh Campus
        - Advising them on how to organize their analytics
    - Working Lexis Nexis on big data and text mining.
        - My army of students found significant improvements to their industrial data mining tools. Also, this team was the first to demonstrate to Lexis Nexis the value of crowdsourcing with Mechanical Turk for building web-scale knowledge bases.
        - In recognition of this,, I was asked to be a panelist at
            - Legal Tech New York, Feb'16
            - The HPCC Summit, Delray Beach, Florida, October'15. Gave a half hour talk to the attendees and served on a panel.
        - Note: for all this work, I received a "outsanding service" award from Lexis Nexis

3. *In the last year, what have you done to help the department:*
    a) *Recruit and retain faculty of the highest caliber.*
        - *Engage with other faculty outside of normal work activities.*
            - Luncheons to discuss research ideas with EE and English Faculty
        - *Search committees*
            - Served on hiring committees for: Kathyryn Stolee and Baishakhi Ray
            - Meet with numerous candidates as part of their on–site interviews. Did not track names this years since this form is new. Will do better next year. But names included Ben Samuel . Brian Magerko, Also had dinner with games candidates e.g. Chris Martens
            - For faculty candidates who received a job offer (e.g. Kathyrn Stolee), I advised on local real estate.
        - *I serve as mentor for*
            - Associate professor Dennis Bahler
            - Associate professor Vince Freeh (meet, several times)
            - Assistant professor Kathryn Stolee (meet, numerous times)

- *Helping junior colleagues with grants*
  - I wrote 80% of the successful NSF REU grant submitted by assistant professor Chris Parnin
  - I am currently co-writing two NSF medium grants with assistant professors Stolee + Parnin
- *Working with other senior colleagues on grants*
  - I co-wrote an NSF small grant with full professor Laurie Williams

- *What research have you contacted Matt Shipman about?*
  - Given numerous presentations at days organized by Prof. Shipman for IBM and CISCO
- *What have you done to promote your press coverage?*
  - Very active on social media

b) Recruit and retain high-quality PhD students.

- The SE faculty are particularly active in Ph.D. recruitment. We inspect the candidates, identify our favorites, then (for all the ones approved by the Grad Office), we engage the students in an extended email conversation. I personally engaged with five such students.
- As to retaining grad students, I've created and maintained extensive relationships with Lexis Nexis and ABB in order to offer students interesting big data projects.
- I also supervise a large number of Ph.D. students (nine)

c) Increase multi-disciplinary research collaborations, including sustainable centers of excellence.

- I am now on the committee for an English digital Humanities Ph.D. candidate (Peter Kudenov)

d) Encourage active collaboration with business, government, and other universities

- I have weekly meetings with Lexis Nexis (Raleigh campus) where my team of students work on their industrial research problems.

e) Expand collaboration and outreach to K-12 students and teachers

- I wrote 80% of the successful NSF REU grant submitted by assistant professor Chris Parnin

f) Offer outstanding and current curricula at the undergraduate and graduate levels

- Internal ABET reviewer, 4th year Capstone subject,

g) Creation or used a repository of course material available to other faculty

- All my course material is on-line, available for download via Github
- E.g. Teaching materials for Model-based Automated Software Engineering http://tiny.cc/mase
- e.g. Complete notes/ project definitions for Graduate SE: https://github.com/txt/se16

h) Incorporated into classes of team projects, written reports, in-class presentations

- All my CSC 510 and CSC 591/791 students write numerous conference-submission-standard essays.
- All my CSC 510 students work in large teams (3 to 5).
- All my CSC 510 students report back to the class in marathon 3 week-long presentation at end of semester.

i) Application for undergraduate research funding

- I wrote 80% of the successful NSF REU grant submitted by assistant professor Chris Parnin

4. In the last year, what challenges/struggles have you had this past year?

- Time management
- Managing grad student expectation. NC State students belong to an international community and those students should spend as much time study that community as studying their won local research at this site.

5. In the last year, what could the department do to make you more productive? (Be specific)

- A budget for Mechanical Turk Experiments
- A better and simpler way to auto-provision HPC for large experiments

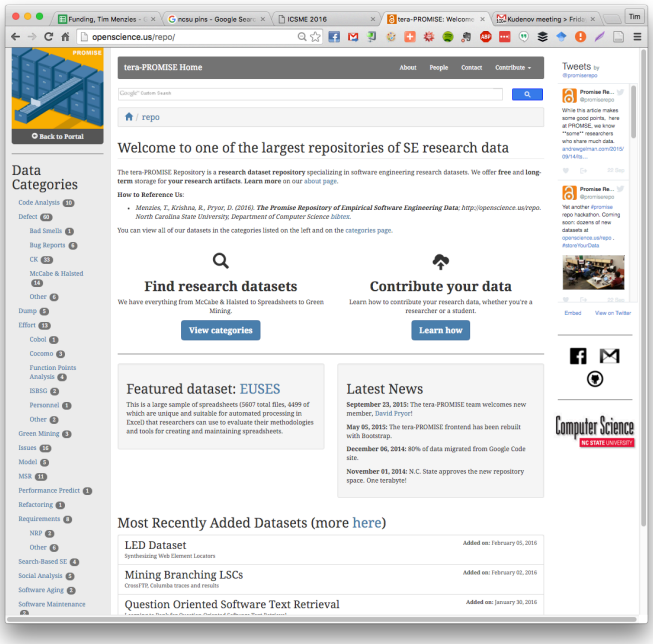
6. What are your goals for the 2016-2017 year?

- More publications for my graduate students
- More long term funding for my graduate students
- One more book on foundations of software science.

7. In prior years, what was your impact over your entire career?

- See next page

B2. IMPACT, full career



The PROMISE Project: *I have been very active in solving an outstanding problem in software engineering: accessing the data required for repeatable experiments.* In order to support the above research, I had to create a source of SE data. Accordingly, in 2005, I founded the PROMISE conference on repeatable experiments in software engineering. The goal of PROMISE is to give the research community free access to the data sets that we can use for analysis of software engineering data.

Optimization of Software-Intensive Systems : *I have developed (and demonstrated the value of) very fast non-numeric optimizers for software-intensive systems.* For the purposes of controlling spacecraft re-entry guidance software, my learners generated better controllers and ran 40 times faster than the state- of-the-art numeric optimizers[3].

- For the purposes of extracting products from product lines, for 2013,2014, my algorithms represented the state of the art on that field (most number of goals, largest models[4]).

Software Defect Prediction: *I have been instrumental in the development of experimental methods that allow for the discovery and precise evalution of software defect predictors generated by data miners from static code attributes.* According to Martin Shepperd my IEEE TSE 2007[5] paper on learning defect predictors "represents the state of the art for this kind of research" and "is widely cited".

- This work is the most cited paper 2005 to 2014 in IEEE Transactions in Software Engineering (source: http://academic.research.microsoft.com).

Cross-company Learning : *I was one of the first to demonstrate that effective local effort/defect models can be learned from data imported from other companies. I* have shown that useful models can be built if *relevancy filtering* selects the subset of the data from other companies that is most relevant to the local company[6].

- This work is the third-most cited paper in the Empirical Software Engineering Journal 2009 to 2014 (source: Google Scholar).

Software Effort Estimation : *I remain one the most prominent publishing researches in this field.* Elsewhere, I have addressed, and reduced, one of the major outstanding problems in the field of effort estimation*; i.e conclusion instability.* Using Ensemble learning*, I* have found that if we study enough data sets and enough learners then the magnitude of that instability is less than the total sample[7].

- Another 2012 paper of mine on analogy and effort estimation as the most-cited estimation paper in the last five years[8] (source: Google Scholar metrics)

---

3 Gregory Gay, Tim Menzies, Misty Davies, and Karen Gundy-Burlet. Automatically finding the control variables for complex system behavior. Automated Software Engineering, (4), December 2010. Available from http: //menzies.us/pdf/10tar34.pdf.

4 Scalable product line configuration: A straw to break the camel's back, ASE , 2013 , AS Sayyad, J Ingram, T Menzies, H Ammar

5 Tim Menzies, Jeremy Greenwald, and Art Frank. Data mining static code attributes to learn defect predictors. IEEE Transactions on Software Engineering, January 2007. Available from http: //menzies.us/pdf/06learnPredict.pdf.

6 B. Turhan, T. Menzies, A. Bener, and J. Distefano. On the relative value of cross-company and within-company data for defect prediction. Empirical Software Engineering, 2009. Available from http://menzies.us/pdf/08ccwc.pdf.

16 On the Value of Ensemble Effort Estimation" by E. Kocaguneli and Tim Menzies and J. Keung. IEEE Transactions on Software Engineering,2011 . 38(6): 1403-1416 (2012)

8 Kocaguneli, E.; Menzies, T.; Keung, J.; Cok, D.; Madachy, R.; , "Active Learning and Effort Estimation: Finding the Essential Content of Software Effort

Software Requirements Engineering: *I was one of the earliest pioneers in the field of search-based software engineering for requirements engineering.* According to Mark Harman, in 2002 I was one of the earliest to apply Pareto optimality in search-based software engineering (SBSE) for requirements engineering[9]

Measurement Errors : *I have discovered a previously undocumented subtle, and dangerous, aspect of a widely-used performance measure.* Precision is a commonly-used assessment measure used in data mining. In 2007, I showed that this seemingly simple measure had significant problems when the target class is relatively rare (specifically, for such data sets, seemingly minor changes in the learning process can lead to massive changes in the precision values)[10].

Estimation Data," Software Engineering, IEEE Transactions on ,

9 M.S. Feather and T. Menzies. Converging on the optimal attainment of requirements. In IEEE Joint Conference On Requirements Engineering ICRE'02 and RE'02, 9-13th September, University of Essen, Germany, 2002. Available from http://menzies.us/pdf/02re02.pdf.

10 Tim Menzies, Alex Dekhtyar, Justin Distefano, and Jeremy Greenwald. Problems with precision. IEEE Transactions on Software Engineering, September 2007. http://menzies.us/pdf/ 07precision.pdf.

VI. SERVICE TO THE UNIVERSITY AND PROFESSIONAL SOCIETIES

A. UNIVERSITY SERVICE

- **Member two search committees, CSC, 2016**
- Member, Two search committees, Computer Science Software Engineering (2015)
- Volunteer, Open Day, March 2015
- Speaker, Graduate research seminar series (CS), November '14

B. NATIONAL AND INTERNATIONAL SERVICE

- Editorial Board
  - **Big Data Research, 2016 -present**
  - **Software Quality Journal, 2016- present**
  - **Information Software Technology, 2016-present**
  - Empirical Software Engineering International Journal, 2009-present
  - Automated Software Engineering journal (2010 – present)
- Associate Editor
  - IEEE Transactions on Software Engineering, 2011-present
- General Chair
  - **IEEE International Conference Software Maintenance and Evolution, 2016**
- Program Chair/Co-Chair:
  - **Symposium Search-Based Software Engineering, 2017**
  - International Conference on Software Engineering, New and Emerging Ideas Track (2015) Florence, Italy.
  - IEEE Automated Software engineering, 2012, Essen, Germany
  - PROMISE conference on repeatable experiments in software engineering (2005-2010)
- Steering Committee Member
  - IEEE Automated Software engineering, 2012-
  - PROMISE conference on repeatable experiments in software engineering (2006-2012)
- Doctoral Symposium
  - Chair, IEEE Automated Software engineering, 2011, Lawrence, Kansas
- Research Proposal Panel
  - National Science Foundation, US (2002, 2004, 2005, 2007, 2009, 2011, 2012, 2007, 2008, 2009, 2010, 2011, 2012, 2012, 2014,2015)
- Guest Editor:
  - **(2016) Automated Software Journal, Best papers RAISE'15**
  - (2015): Automated Software Journal, Best papers, ASE conference, 2011-2012
  - (2015) Special issue, best papers from RAISE'13, Automated Software Engineering
  - (2013) Two special issues, IEEE Software, Software Analytics (with Thomas Zimmermann).
  - (2013) Special Issues, Information and Software Technology, Best papers from PROMISE'11, 55(8),.
  - (2013): Special Issue, Empirical Software Engineering, Best papers, PROMISE'10, 18(3) 2013
  - (2012) Special Issue, Automated Software Engineering, "Learning to Organize Testing", 19(2), 2012.
  - (2012): Special Issue, Empirical Software Engineering, Jan 2012, "Conclusion Stability in SE"
  - (2012): Special Issue, Best papers RAISE 2012, Software Quality Journal
  - (2010): Special issue: Automated Software Engineering, Repeatable Experiments in Effort Estimation",;
  - (2009): Special issue: Journal of Empirical Software Engineering, ' "IR for Program Comprehension", 2009;
  - (2008) :Special issue: Journal of Empirical Software Engineering, ' "Repeatable Experiments in SE",
  - (2003) :Special issue, Requirements Engineering Journal, "Model-based requirements engineering
  - (2003): Special issue of IEEE Intelligent Systems, "AI's Second Century", 2003.
  - (1999, 1998): Two special issues of International Journal of Human Computer Studies (IJHCS),
- General chair
  - **ICSME'16**
  - **BigDSE'16**, BigBDSE'15
- Senior roles in conference organization:
  - **PC-chair: SSBSE 2017,**
  - **Artifacts chaor: FSE 2016**
  - **Artifacts chair: ICSME 2016**

- Program Committee:
  - **2016:**
    - **ASE'16, BIGDSE'16, EASE 2016, ESEM2016, ICSE-SRC 2016, ISSRE 2016, PROMISE 2016, RAISE 2016, SCORE 2016**
  - 2015:
    - Ase'15, BigDSE'15, Ease'15, EsPreSSE'15, Esem'15, Fse'15, Gecco'15, Icpc'15, Issre'15, Msr'15, NasBase'15, Promise'15, Raise'15, Ssbse'15
  - 2014:
    - MSR'14, ICSE14-demos, ICSE14-mainConference, DAPSE'14, EASE'14, GTSE'14, SAM 2014, SEAA 2014,
  - Before 2014:
    - Mining Software Engineering 2013, 2012, '2011
    - IEEE Automated Software Engineering (2013,2012,2011,2010,2009, 2008,2007,2005, 2004, 2003, 2002)
    - Empirical Software Engineering and Measurement '2012 '2011, 2013
    - SAM2103,
    - DAPSE'13
    - ICSE'13: demos
    - ASE-Tools'13
    - ISSRE'13
    - GTSE'13
    - MALIR'13
    - Software Mining -2012, 2013
    - RAISE'12, RAISE'13
    - FSE New ideas'11,
    - Software engineering week, 2011,
    - Spark'11
    - IEEE International Symposium on Software Reliability Engineering (2010,2009);
    - Pacific Knowledge Acquisition Workshop, 2009,2008
    - LSO (learning software organizations), 2008
    - Traceability in Emerging forms of SE , 2007
    - International Workshop on Living with Uncertainty (2007)
    - IEEE conference on high assurance software engineering (2007, 2004);
    - 17th International Conference on Automated Planning & Scheduling (2007)
    - MoChArt '05 (model checking and AI)
    - Tim Menzies, vita page 7 of 23
    - IEEE NASA Software Engineering Workshop (2003)
    - IEEE Metrics 2003;
    - Numerous other PCs since 1991 including
      - 8 international conferences
      - 16 international workshops,
      - 5 Australian national workshops.
      - Organizing committee member for 2 international workshops, 4 national conferences and workshops.
- Reviewer for:
  - ACM Transactions on Software Engineering and Methodology, IEEE Transactions on Software Engineering, Empirical Software Engineering, Automate Software Engineering, Information Systems and Technology, Applied Soft Computing, IEEE Software, International Journal of Human Computer Studies. Software Quality Journal, Software Process: Improvement and Practice Journal, Software Testing, Verification, and Reliability