
Curriculum Vitae
Tim Menzies
Sept 2021

tjmenzie@csc.ncsu.edu; timmm@ieee.org
URL: <http://menzies.us>

BRIEF RESUME (for details, see below)

Education background:

- Ph.D., CS, University of New South Wales, 1995 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, 1985.

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

Funding:

- \$13,000,000+ from NSG, Facebook, NASA, National Archives, LexisNexis Grammatech, IBM and other sources

Awards:

- *IEEE Fellow* (2019),
- Mining Software Repositories Foundational Contribution Award, 2017
- Google scholar ranks me, internationally, as #1 in search-based software engineering, #1 In defect prediction; #2 in software cost estimation, #2 in configuration, #3 in software analytics

Publications:

- *H-index*: 62 (Aug'21).
- *Papers*: 86 journal + 129 conference + 4 books + 86 other
- Author of one of SE's 20-most cited papers (cites/year);
- ACM Sigsoft distinguished paper award ICSE'19 and FSE'2,
- Ten year most influential paper award, ICSME'19 (for an ICSME'09 paper)

Government work:

- *NASA software research chair*: 2002-2004
- *NSF panelist*: 13 times (2003-2020)

Journal work:

- *Editor in Chief*: *Automated Software Engineering*
- *Assoc. ed.*: CACM, TSE, TOSEM, JSS, EMSE, IST, ASEJ, IEEE Software, SQJ, Big Data Research, IET Software

Conference work:

- *co-general chair*: ICMSE'16; RAISE'19, PROMISE'05..'12
- *co-PC chair*: PROMISE'20, SSBSE'17, NEIR'15, ASE'12
- *artifacts co-chair*: ICSE'20, ASE'20, FSE'18, FSE'16
- *program committees*: ASE'20, ICSE'20, ESEM'20, FSE'19, ASE'19, MSR'19, SSBSE, PROMISE,...

Research students: (total)

- Ph.D.: 12 current. 12 past. Masters (by research): 31
- Current graduate students (at the RAISE lab: real-world AI for SE

Membership in professional organizations:

- Association for Computing Machinery (ACM), 1996-present
- Institute of Electrical and Electronic Engineers (IEEE), 1997-present
- Promotion to IEEE senior member in 2017
- Elevated to IEEE Fellow, 2019.

Scholarly and professional honors:

1. ACM Sigsoft Distinguished paper award, FSE'21 Bias in Machine Learning Software: Why? How? What to Do?
2. ACM Sigsoft Distinguished paper award, ICSE'19, "iSENSE: Completion-Aware Crowdttesting Management"
3. Most Influential Paper Award (from ICSM 2009) "On the use of Relevance Feedback in IR-based Concept Location"
4. ACM TOSEM journal distinguished reviewer 2011-2018 (only person to receive that award for all those years)
5. Journal of Software & Systems : best reviewers for 2018 (only 12 such people selected).
6. Inaugural Mining Software Repositories Foundational Contribution Award., 2017
7. Carol Miller Graduate Lecturer Award, Association for Computing Machinery/Association of Information Technology Professionals, 2016
8. Distinguished reviewer, ACM Transactions on Software Engineering Methodologies, 2016
9. Outstanding reviewer award, journal of Information and Software Technology, 2016
10. Service award from Big Data community: Lexis Nexis, 2015
11. Distinguished reviewer, ACM Transactions on SE Methodologies, 2015
12. WVU College of Engineering, Outstanding Researcher, 2010
13. NASA Commendation for Chief of Mission Assurance, 2004

Professional service on campus:

- Organizer Carla Savage awards, NCSU Computer Science
- Proxy to the chair at college meetings
- Chair, faculty search committee
- Member, departmental graduate review committee
- NC State Member , CSC Faculty Search (2015, 2016,2017)
- Curating the PROMISE repository of SE data. That work recently won me the inaugural Mining Software Repositories Foundational Contribution Award. See <http://2017.msrrconf.org/#/awards>
- NC State Member , Software Engineering Faculty Search (2014)
- NC State, Open house weekend (March 2015)
- 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)

Professional service off campus :

- Editor in chief, Journal of Automated Software Engineering
- Artifacts co-chair, ICSE'20, ASE'21
- Three SIGSOFT standards committee: artifacts, data science, search-based optimization
- ROSE festival organizer, FSE'18, ICSE'19, ICSE'20, FSE'20
- Co-PC chair PROMISE'20
- PC-chair RAISE'19
- Artifacts chair, FSE'18
- Co-PC chair, SSBSE'17
- Co-chair, SWAN'17
- Co-General Chair: International Conference on Software Maintenance and Evolution 2016
- Co-Program Chair: PROMISE'20, SSBSE'17, ICSE NIER'15, ASE'12.
- Associated editor: ACM Transactions on Software Engineering
- Associated Editor: Information Software Technology
- Editorial Board: Software Quality Journal
- Editorial Board: Big Data Research Journal.
- Chair, IEEE Software editor review board, 2016
- Associate Editor: IEEE Transactions on Software Engineering 2011-2017.
- Editorial Board: Empirical Software Engineering International Journal, 2009-present; Automated Software Engineering Journal, 2010-present
- Steering Committee Member:
 - IEEE Automated Software Engineering (2012-present)
 - PROMISE conference 2005-2012.
- PC member:
- IJCAI'21, AAAI'21, MSR'21, ESEM'21, ASE'20, ESEM'20, ICSE20, MSR/19, FSE'19, ICSE'20 ICSE,18, IEEE
- Distinguished paper committee: MSR'20
- IEEE Fellow award committee (2020)
- ICSE'20 program committee
- IEEE Technical committee, Software Engineering, Member at large
- Msr award committee 2018, SSBSE'18, ESEM 2018, SoftwareMining'17, ICSE'17 (artifacts), PROMISE'18, MSR'18, ESEM'18, SPLC'17, EASE'17 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, Icp'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 (2002-2014), 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.

TEACHING AND MENTORING

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

- Graduate software engineering 2020, 2021 cs510.
- 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

INSTRUCTIONAL DEVELOPMENT

- New course Foundations of Software Science: At NC State in 2017 I created and taught the new .All lectures and projects were written by me.
- *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.

MENTORING ACTIVITIES

- Working with my SE faculty colleagues, developed a successful NSF Research Experience for Undergraduates grant for “Science of Software”, 2016-2018.
- I currently serve as mentor for assistant professor Tom Price, Bradley Reaves and Associate professor Vince Freeh, Khaled Harfoush, Sarah Heckman

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 + 279MS):

Student working towards a degree (12 PhD):

1. Xiao Ling
2. Rahul Yedida
3. Kewen Peng
4. Tianpei (Patrick) Xia
5. Huy (Ken) Tu
6. Suvodeep Majumder

7. Joymallya Chakraborty
8. Rui Shu
9. Shrikanth Chandrasekaran
10. Xueqi (Sherry) Yang

Completed Ph.D.:

1. Zhe Yu (2020, Total Recall and Software Engineering)
2. Rahul Krishna (2019, Learning Actionable Analytics in Software Engineering)
3. Amritanshu Agrawal (2019, On the Nature of Software Engineering Data)
4. Jianfeng Chen (2019, On the Value of Sampling and Pruning for Search-Based SE)
5. Vivek Nair, (2019, NCSU) Frugal Ways to Find Good Configurations.
6. Wei Fu, (2018, NCSU) Simpler Software Analytics: When? When Not?
7. Abdel Sayyad Ph.D. (2014, WVU) *Evolutionary Search Techniques with Strong Heuristics for Multi-Objective Feature Selection in Software Product Lines*
8. Joe Krall Ph.D. (2014, WVU) *Active Learning for Search-Based Software Engineering*
9. Fayola Peters Ph.D. (2014, WVU) *Privacy and Data Sharing*
10. Ekrem Ph.D. (2012, WVU) *A Principled Methodology: A Dozen Principles of Software Effort Estimation*
11. Nandeshwar, Ashutosh Ph.D. (2011, WVU) *Longitudinal study of first-time freshmen using data mining*
12. David Owen Ph.D. (2010, WVU) *Combining complementary formal verification strategies to improve performance and accuracy*
13. Scott Chen Ph.D. (2004, U.Sc.) *Data Mining for Effort Estimation*

Completed Masters (where I was committee chair)

1. FNU Vivek, MS, NC State, 2019
2. Akshay Nalwaya MS, NC State, 2019
3. Sushma Ravichandran, MS, NC State, 2018
4. George Mathew, MS, NC State, 2016
5. Rahul Krishna, MS, NC State, 2016
6. Divya Ganesan MS (2015, WVU) *Exploring Essential Content of Defect Prediction and Effort Estimation through Data Reduction*
7. Naveen Kimar, WVU, 2014
8. Ben Province MS (2015, WVU), *The Effects of Parameter Tuning on Machine Learning Performance in a Software Defect Prediction Context.*
9. Vasil Papakroni MS (2013, WVU) *Data Carving: Identifying and Removing Irrelevancies*
10. Joseph Craig MS (2013, WVU) *Accelerating MOEA Non-dominated Sorting by Preserving Archival Relationships*
11. Will Burney MS (2012, WVU) *Understanding Search-Based Software Engineering*
12. Adam Brady MS (2011, WVU) *W2 : a simple, flexible, case-based recommendation engine for software quality*
13. Brian Lemon MS (2010, WVU) *The effect of locality based learning on software defect prediction*
14. Fayola Peters MS (2010, WVU) *CLIFF: finding prototypes for nearest neighbor algorithms with applications*
15. Andrew Matheny MS (2010, WVU) *Trade-offs of heuristic vs. rigorous algorithms in text mining*
16. Joe D'alessandro MS (2010, WVU) *Optimized trusted information sharing*
17. Grey Gay MS (2010, WVU) *The robust optimization of non-linear requirements models*
18. Adam Nelson MS (2010, WVU) *OURMINE: an open source data mining toolkit*
19. Ous El-waras MS (2008, WVU) *Software process control without calibration*
20. Omid Jalali MS (2008, WVU) *Evaluation bias in effort estimation*
21. Zach Milton MS (2008, WVU) *Which: a stochastic best-first search learner*
22. Brian Sower MS (2008, WVU) *Increasing the performance and realism of procedurally generated buildings*
23. Justin DiStefano MS (2008, WVU) *Building better software : the applicability of a professional tool for automata*
24. Daniel Baker MS (2007, WVU) *Hybrid approach to expert and model based effort estimation*
25. Donald Boland MS (2007, WVU) *Data discretization simplified: randomized binary search trees for data preprocessing*
26. Jeremy Greenwald MS (2006, Portland State) *Understanding procedural Knowledge*
27. Ryan Clark MS (2005, Portland State) *Optimizing Treatment Learning*
28. Kareem Ammar (2004, WVU) *Multi-heuristic theory assessment with iterative selection* [
29. Yi Hu MS (2003, University British Columbia) *Treatment learning*
30. Eliza Chaing MS (2003, University British Columbia) *Early LifeCycle Simulation of Software Process Models.*
31. David Owen MS (2002, WVU) *Combining complementary formal verification strategies to improve performance and accuracy*
32. John Powell MS (2001, WVU) *Graph theoretic approach to assessing tradeoffs on memory usage for model checking*

SCHOLARSHIP IN THE REALMS OF FACULTY RESPONSIBILITY

SCHOLARLY ACCOMPLISHMENTS – PUBLICATIONS

Invited and Contributed Research Presentations

1. Keynote ICSME, 2020
2. Keynote, ICSA, 2020
3. ISR Distinguished Speaker, UCI, Series 2020.
4. Two journal first presentations, ICSE'20
5. Invited Task, ISR, UCLA, 2020
6. Tutorial, LASER summer school on SE (Italy)
7. Keynote, ICSE 2019 SEIP
8. One journal first presentations, ICSE'19
9. Two journal first presentations FSE'19
10. Invited Talk, CodeFreeze19, Minnesota, 2019
11. Invited keynote, Foundations of SE, Florida, 2018
12. Invited keynote, ICSE RAISE workshop, Realizing AI+SE synergies, , May 2018
13. Invited Keynote, Inaugural MSR award, MSR'17
14. Invited Keynote , Data-Driven Search-Based SE, Japan, December 2017.
15. Keynote: SWAN'16 , Seeking simpler software analytics : <http://tiny.cc/timm13>
16. Keynote NSF PI meeting on sustainable development,, Feb 2017 <http://tiny.cc/nsf17>
17. Keynote SBST'16 (at ICSE'16): Testing: the (w)hole story.
18. Keynote, ICSE'15 workshop keynote (WetSOM'14): What Metrics matter. Hyderabad, India.
19. Invited Talk, Naval Research Lab, May 2018
20. Invited Talk, Lexis Nexis Cognitive Summit, August 2018
21. Invited Talk, Monash University Dean's series, Australia, July 2017
22. Invited Talk, IBM Technical Interchange Conferences, November 2017
23. Invited ICSE talk: Trends in topics at SE conferences (1993-2013). ICSE (Companion Volume) 2017: 397-398
24. Invited talk, 52nd CREST workshop, University College, London. <http://tiny.cc/timcow52>
25. Invited speaker, Lexis Nexis industry day, August 2016 <http://tiny.cc/timm9>
26. Journal-First presentation, FSE'17 Are Delayed Issues Harder to Resolve?
27. Journal-First presentation, FSE'15 Geometric Active Learning
28. Tutorial, ICSE'16: How Not to do it: Anti-patterns in data science
29. Tutorial, ICSE'15: Art and Science of Analyzing Software Data

Refereed Journal and Top Magazine Publications

NCSU Ph.D. students are marked in **bold**.

My own Ph.D. students are marked in **bold underline**. Note: my students are publishing at very senior venues.

1. **Xiao Ling, Rishabh Agrawal**, and Tim Menzies, How Different is Test Case Prioritization for Open and Closed Source Projects? IEEE Transactions on Software Engineering. Accepted for publication Feb 2021
2. **Shrikanth N.C.**, William Nichols, **Fahmid Morshed Fahid** · Tim Menzies Assessing Practitioner Beliefs about Software Engineering, EMSE journal, accepted for publication Jan 2021
3. Anita D. Carleton, Erin Harper, Michael R. Lyu, Sigrid Eldh, Tao Xie, Tim Menzies, Expert Perspectives on AI. IEEE Softw. 37(4): 87-94 (2020)
4. Anita D. Carleton, Erin Harper, Tim Menzies, Tao Xie, Sigrid Eldh, Michael R. Lyu: The AI Effect: Working at the Intersection of AI and SE. IEEE Softw. 37(4): 26-35 (2020)
5. Junjie Wang, Ye Yang, Tim Menzies, Qing Wang: iSENSE2.0: Improving Completion-aware Crowdstesting Management with Duplicate Tagger and Sanity Checker. ACM Trans. Softw. Eng. Methodol. 29(4): 24:1-24:27 (2020)
6. **Kewen Peng**, Tim Menzies Defect Reduction Planning (using TimeLIME), IEEE Transactions on Software Engineering, accepted, to appear
7. **R Krishna**, T Menzies, From Prediction to Planning: Improving Software Quality with BELLTRE, Empirical Software Engineering, 2020
8. **R Shu, T Xia, J Chen**, L Williams, T Menzies, Improved Recognition of Security Bugs via Dual Hyperparameter Optimization, Empirical Software Engineering, 2020.

9. **Rahul Krishna**, Tim Menzies: Learning actionable analytics from multiple software projects. *Empir. Softw. Eng.* 25(5): 3468-3500 (2020)
10. **T Xia, R Shu**, X Shen, T Menzies, Sequential Model Optimization for Software Effort Estimation, *IEEE Transactions on Software Engineering*, 2020
11. Tim Menzies: The Five Laws of SE for AI. *IEEE Softw.* 37(1): 81-85 (2020)
12. **Tu, Huy, Zhe Yu**, and Tim Menzies, "Better data labelling with EMBLEM (and how that impacts defect prediction)." *IEEE Transactions on Software Engineering* (2020).
13. **X Yang, J Chen, R Yedida, Z Yu**, T Menzies, Learning to Recognize Actionable Static Code Warnings (is Intrinsically Easy), *Empirical Software Engineering* 2020
14. **X Yang, Z Yu**, J Wang, T Menzies, Understanding static code warnings: An incremental AI approach, *Expert Systems with Applications*, 114-134, 2020
15. **Z Yu, FM Fahid, H Tu**, T Menzies , Identifying Self-Admitted Technical Debts with Jitterbug: A Two-Step Approach *IEEE Transactions on Software Engineering*, 2020
16. **Z Yu, C Theisen**, L Williams, T Menzies , Improving Vulnerability Inspection Efficiency Using Active Learning. *IEEE Transactions on Software Engineering*, 2019.
17. **A Agrawal, W Fu, D Chen**, X Shen, T Menzies. How to" DODGE" Complex Software Analytics Accepted to appear *IEEE Transactions on Software Engineering*, 2019.
18. **Zhe Yu, Christopher Theisen** ; Laurie Williams ; Tim Menzies, Improving Vulnerability Inspection Efficiency Using Active Learning, to appear *IEEE Transactions of SE*, 2019.
19. Junjie Wang ; Song Wang ; **Jianfeng Chen** ; Tim Menzies ; Qiang Cui ; Miao Xie ; Qing Wang, Characterizing Crowds to Better Optimize Worker Recommendation in Crowdsourced Testing, *IEEE Transactions on SE*, to appear, 2019
20. **Amritanshu Agrawal**, Tim Menzies, Leandro L. Minku, Markus Wagner, **Zhe Yu**, Better Software Analytics via "DUO": Data Mining Algorithms Using/Used-by Optimizers. , *Journal of Empirical Software Engineering*, 2019
21. Junjie Wang, Mingyang Li, Song Wang, Tim Menzies, Qing Wang: Images don't lie: Duplicate crowdtesting reports detection with screenshot information. *Information & Software Technology* 110: 139-155 (2019)
22. Tim Menzies, Martin J. Shepperd: "Bad smells" in software analytics papers .*Information & Software Technology* 112: 35-47 (2019)
23. **Rahul Krishna**, Tim Menzies: Bellwethers: A Baseline Method for Transfer Learning. *IEEE Trans. Software Eng.* 45(11): 1081-1105 (2019)
24. **Vivek Nair, Zhe Yu**, Tim Menzies, Norbert Siegmund, Sven Apel, Finding Faster Configurations using FLASH, *IEEE Transactions on SE*, 2018
25. **G Mathew, A Agrawal**, T Menzies. Finding Trends in Software Research, *IEEE Transactions on Software Engineering*, 2018.
26. M Choetkiertikul, HK Dam, T Tran, TTM Pham, A Ghose, T Menzies , A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*
27. **Zhe Yu**, Nicholas A. Kraft, Tim Menzies, Finding better active learners for faster literature reviews. *Empirical Software Engineering* 23(6): 3161-3186 (2018)
28. **Amritanshu Agrawal, Wei Fu**, Tim Menzies: What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information & Software Technology* 98: 74-88 (2018)
29. **Z Yu**, T Menzies. , FAST2: An intelligent assistant for finding relevant papers *Expert Systems with Applications* 120, 57-71, 2019
30. J Nam, **W Fu**, S Kim, T Menzies, L Tan, Heterogeneous defect prediction, *IEEE Transactions on Software Engineering*, June 2017
31. **Jianfeng Chen, Vivek Nair, Rahul Krishna**, Tim Menzies:"Sampling" as a Baseline Optimizer for Search-Based Software Engineering. *IEEE Trans. Software Eng.* 45(6): 597-614 (2019)
32. **Vivek Nair**, Tim Menzies, Norbert Siegmund, Sven Apel: Faster discovery of faster system configurations with spectral learning. *Autom. Softw. Eng.* 25(2): 247-277 (2018)
33. **Jianfeng Chen, Vivek Nair**, Tim Menzies: Beyond evolutionary algorithms for search-based software engineering. *Information & Software Technology* 95: 281-294 (2018).
34. Rahul Krishna, Tim Menzies, Lucas Layman: Less is more: Minimizing code reorganization using XTREE. *Information & Software Technology* 88: 53-66 (2017)
35. R Pandita, R Jetley, S Sudarsan, T Menzies, L Williams, TMAP: Discovering relevant API methods through text mining of API documentation, *Journal of Software: Evolution and Process*, accepted 2017. .
36. T Menzies, Y Yang, G Mathew, B Boehm, J Hihn, , Negative results for software effort estimation *Empirical Software Engineering*, 1-26, 3, 2016
37. T Menzies, N William, F Schull, L Layman Are Delayed Issues Harder to Resolve? Revisiting Cost-to-Fix of Defects throughout the Lifecycle *Empirical Software Engineering* . Issue 4/2017.
38. W. Fu, T Menzies, X. Shen, Tuning for Software Analytics: is it Really Necessary? *Information and Software*

Technology, Volume 76, August 2016, Pages 135-146

39. 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.
40. 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
41. Transfer learning in effort estimation, E Kocaguneli, T Menzies, E Mendes *Empirical Software Engineering*, 1-31, 2014
42. SN Partington, V Papakroni, T Menzies , Optimizing data collection for public health decisions: a data mining approach, *BMC Public Health* 14 (1), 593, 2014
43. 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
44. 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
45. 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
46. 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)
47. 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)
48. 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), 1698-1713, 2013
49. 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)
50. "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
51. 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* ,
52. 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,
53. 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)
54. 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)
55. "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)
56. 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
57. 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
58. 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.
59. "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 .
60. "What is Enough Quality for Data Repositories?" by Tim Menzies and Adam Brady and Ekrem Kocaguneli. *Software Quality Professional*, volume 13, number 3, 2011 .
61. 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> .
62. 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> .
63. 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> .
64. Tim Menzies and Omid Jalali and Jaius 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> .
65. 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> .
 66. Oussama El-Rawas and Tim Menzies, "A Second Look at Faster, Better, Cheaper" in. Innovations Systems and Software Engineering pages 319-335 2010 .
 67. 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> .
 68. 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> .
 69. 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> .
 70. 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>
 71. 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.
 72. 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>.
 73. T. Menzies, M. Benson, K. Costello, C. Moats, M. Northey, and J. Richardson. Learning better IV&V practices. Innovations in Systems and Software Engineering, March 2008. Available from <http://menzies.us/pdf/07ivv.pdf>.
 74. 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>.
 75. 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>.
 76. 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>.
 77. 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>.
 78. T. Menzies, D. Owen, and J. Richardson. The strangest thing about software. IEEE Computer, 2007. <http://menzies.us/pdf/07strange.pdf>.
 79. Tim Menzies, Zhihao Chen, Jaius 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>.
 80. T. Menzies and J. Richardson. Making sense of requirements, sooner. IEEE Computer, October 2006. Available from <http://menzies.us/pdf/06qrre.pdf>.
 81. T. 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>.
 82. T. Menzies and C. Pecheur. Verification and Validation and Artificial Intelligence. In M. Zelkowitz, editor, Advances in Computing, volume 65. Elsevier, 2005. Available from <http://menzies.us/pdf/04aivv.pdf>.
 83. T. Menzies, R. Gunalan, 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>.
 84. Zhihao Chen, Tim Menzies, Dan Port, and Barry Boehm. Finding the right data for software cost modeling. IEEE Software, Nov 2005.
 85. 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>.
 86. 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>.
 87. T. Menzies and Y. Hu. Data mining for very busy people. In IEEE Computer, November 2003. Available from <http://menzies.us/pdf/03tar2.pdf>.
 88. 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>.
 89. T. Menzies and B. Cukic. When to test less. IEEE Software, 17(5):107–112, 2000. Available from <http://menzies.us/pdf/00iesoft.pdf>.
 90. 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>.
91. 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>.
 92. 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>.
 93. 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>.
 94. T. Menzies. Knowledge maintenance: The state of the art. *The Knowledge Engineering Review*, 14(1):1–46, 1999. Available from <http://menzies.us/pdf/97kmal.pdf>.
 95. T. Menzies. Cost benefits of ontologies. *ACM SIGART Intelligence magazine*, Fall 1999. Available from <http://menzies.us/pdf/99sigart.pdf>.
 96. 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>.
 97. 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>.
 98. 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>.
 99. 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>.
 100. T.J. Menzies. An investigation of the ai and expert systems literature 1980-1984. *AI Magazine*, Summer 1989.
 101. T.J. Menzies. Domain-specific knowledge representations. *AI Expert*, Summer 1989.

Books/Book Chapters

1. Tim Menzies, Justyna Petke: Search Based Software Engineering - 9th International Symposium, SSBSE 2017, Paderborn, Germany, September 9-11, 2017, Proceedings. *Lecture Notes in Computer Science 10452*, Springer 2017, ISBN 978-3-319-66298-5
2. Perspectives on Data Science for Software Engineering, T. Menzies, L. Williams, T. Zimmermann, Morgan Kaufmann, 2016
3. The Art and Science of Analyzing Software Data, C. Bird, T. Menzies, T. Zimmermann, Morgan Kaufmann, 2015
4. Sharing Data and Models in Software Engineering, T. Menzies, Ekrem Kocaguneli, L. Minku, F. Peters, B. Turhan, Morgan Kaufmann, 2014
5. Occam's Razor and Simple Software Project Management T Menzies Software Project Management in a Changing World, 447-472, 2014
6. Data mining: a tutorial T Menzies, Recommendation Systems in Software Engineering. Springer, Berlin, 2014
7. "The Quest for Convincing Evidence" by Tim Menzies and Forrest Shull. *Making Software: What Really Works and We We Believe it* 2010
8. "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> .
9. "Many Maybes Mean (Mostly) the Same Thing" by T. Menzies and H. Singh. *Soft Computing in Software Engineering 2003* . Available from <http://menzies.us/pdf/03maybe.pdf> .
10. "How Many Tests are Enough?" by T.J. Menzies and B. Cukic. *Handbook of Software Engineering and Knowledge Engineering*, Volume II 2002 . Available from <http://menzies.us/pdf/00ntests.pdf> .
11. "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>

12. "Knowledge Elicitation: the State of the Art" by T.J. Menzies. Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available from <http://menzies.us/pdf/00getknow.pdf> .
13. "Evaluation Issues for Visual Programming Languages" by T. Menzies. Handbook of Software Engineering and Knowledge Engineering, Volume II 2002 . Available from <http://menzies.us/pdf/00vp.pdf>
14. "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> .
15. "Expert Systems Maintenance" by T.J. Menzies and J. Debenham. Encyclopedia of Computer Science and Technology pages 35-54 2000 . Available from <http://menzies.us/pdf/00cst.pdf> .
16. "Software Visualization" by P. Haynes and T. Menzies and R.F. Cohen. 1997 . Available from <http://menzies.us/pdf/oovis95.pdf> .

Refereed Conference Publications

NCSU Ph.D. students are marked in **bold**.

My own Ph.D. students are marked in **bold underline**. Note: my students are publishing at very senior venues.

1. **NC Shrikanth, S Majumder**, T Menzies, Early Life Cycle Software Defect Prediction. Why? How? , ICSE'21
2. **J Chakraborty, S Majumder, Z Yu**, T Menzies, Fairway: A Way to Build Fair ML Software. ESEC/FSE 2020
3. **J Chakraborty, K Peng**, T Menzies, Making Fair ML Software using Trustworthy Explanation, International Conference on Automated Software Engineering, 2020
4. **NC Shrikanth**, T Menzies, What disconnects practitioner belief and empirical evidence?, ICSE-SEIP'20.
5. Tim Menzies: Take control: on the unreasonable effectiveness of software analytics. ICSE (SEIP) 2019: 265-266
6. Junjie Wang, Ye Yang, **Rahul Krishna**, Tim Menzies, Qing Wang: iSENSE: completion-aware crowdtesting management. ICSE 2019: 912-923
7. **Di Chen**, Kathryn T. Stolee, Tim Menzies: Replication can improve prior results: a GitHub study of pull request acceptance. ICPC 2019: 179-190
8. **Zhe Yu, Fahmid M. Fahid**, Tim Menzies, Gregg Rothermel, Kyle Patrick, Snehith Cherian: TERMINATOR: better automated UI test case prioritization. ESEC/SIGSOFT FSE 2019: 883-894
9. **Jianfeng Chen, Joymallya Chakraborty**, Philip Clark, Kevin Haverlock, Snehith Cherian, Tim Menzies: Predicting breakdowns in cloud services (with SPIKE). ESEC/SIGSOFT FSE 2019: 916-924
10. **Amritanshu Agrawal**, Tim Menzies: Is "better data" better than "better data miners"?: on the benefits of tuning SMOTE for defect prediction. ICSE 2018: 1050-1061
11. **Di Chen, Wei Fu, Rahul Krishna**, Tim Menzies: Applications of psychological science for actionable analytics. ESEC/SIGSOFT FSE 2018: 456-467
12. **Amritanshu Agrawal**, Akond Rahman, **Rahul Krishna**, Alexander Sobran, Tim Menzies: We don't need another hero?: the impact of "heroes" on software development. ICSE (SEIP) 2018: 245-253
13. **Rahul Krishna, Amritanshu Agrawal**, Akond Rahman, Alexander Sobran, Tim Menzies: What is the connection between issues, bugs, and enhancements?: lessons learned from 800+ software projects. ICSE (SEIP) 2018: 306-315
14. **Suvodeep Majumder**, Nikhila Balaji, Katie Brey, **Wei Fu**, Tim Menzies: 500+ times faster than deep learning: a case study exploring faster methods for text mining stackoverflow. MSR 2018: 554-563
15. **Vivek Nair, Amritanshu Agrawal, Jianfeng Chen, Wei Fu, George Mathew**, Tim Menzies, Leandro L. Minku, Markus Wagner, **Zhe Yu**: Data-driven search-based software engineering. MSR 2018: 341-352
16. Jianfeng Chen, Tim Menzies: RIOT: A Stochastic-Based Method for Workflow Scheduling in the Cloud. IEEE CLOUD 2018: 318-325
17. **Chin-Jung Hsu, Vivek Nair**, Tim Menzies, Vincent W. Freeh: Micky: A Cheaper Alternative for Selecting Cloud Instances. IEEE CLOUD 2018: 409-416
18. Chin-Jung Hsu, Vivek Nair, Vincent W. Freeh, Tim Menzies: Arrow: Low-Level Augmented Bayesian Optimization for Finding the Best Cloud VM. ICDCS 2018: 660-670
19. Wei Fu, Tim Menzies: Revisiting unsupervised learning for defect prediction. ESEC/SIGSOFT FSE 2017: 72-83
20. Vivek Nair, Tim Menzies, Norbert Siegmund, Sven Apel: Using bad learners to find good configurations. ESEC/SIGSOFT FSE 2017: 257-267
21. George Mathew, Tim Menzies, Neil A. Ernst, John Klein: "SHORT"er Reasoning About Larger Requirements Models. RE'17: 154-163
22. J Hihn, M Saing, E Huntington, J Johnson, T Menzies, G Mathew, The NASA analogy software cost model: A web-based cost analysis tool, IEEE Aerospace Conference, 2017, 1-17
23. V Nair, T Menzies, J Chen, An (accidental) exploration of alternatives to evolutionary algorithms for SBSE International Symposium on Search Based Software Engineering, 96-111, 2016.
24. R Krishna, T Menzies, W Fu, Too much automation? the bellwether effect and its implications for transfer learning, ASE'16.

25. J Hihn, L Juster, J Johnson, T Menzies, G Michael, Improving and expanding NASA software cost estimation methods Aerospace Conference, 2016 IEEE, 1-12
26. 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)
27. Jairus Hihn, Tim Menzies, Improving and Expanding NASA Software Cost Estimation Methods 2016 IEEE Aerospace Conference.
28. Scalable product line configuration: A straw to break the camel's back, ASE , 2013 , AS Sayyad, J Ingram, T Menzies, H Ammar
29. 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
30. Class level fault prediction using software clustering, G Scanniello, C Gravino, A Marcus, T Menzies, ASE 2013
31. 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
32. Tim Menzies: Beyond data mining; towards "idea engineering". PROMISE 2013: 11 Learning from open-source projects: An empirical study on defect prediction, Z He, F Peters, T Menzies, Y Yang, ESEM 2013
33. Ekrem Kocaguneli, Bojan Cukic, Tim Menzies, Huihua Lu: Building a second opinion: learning cross-company data. PROMISE 2013: 12
34. Beyond data mining; towards idea engineering T Menzies, PROMISE 2013
35. Ekrem Kocaguneli, Thomas Zimmermann, Christian Bird, Nachiappan Nagappan, Tim Menzies: Distributed development considered harmful? ICSE 2013: 882-890
36. Fayola Peters, Tim Menzies: Privacy and utility for defect prediction: Experiments with MORPH. ICSE 2012: 189-199
37. Yang Sok Kim, Byeong Ho Kang, Seung Hwan Ryu, Paul Compton, Soyeon Caren Han, Tim Menzies: Crowd-Sourced Knowledge Bases. PKAW 2012: 258-271
38. Raymond Borges, Tim Menzies: Learning to change projects. PROMISE 2012: 11-18
39. 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
40. "How to Find Relevant Data for Effort Estimation?" by Kocaguneli, E. and Menzies, T..Proceedings ESEM11, 2011
41. "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> .
42. "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/10extrisk.pdf> .
43. "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>.
44. "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> .
45. "Software is Data Too" by A. Marcus and T. Menzies. FoSER 2010 November 2010 . Available from <http://menzies.us/pdf/10softwareisdata.pdf> .
46. "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> .
47. "Regularities in Learning Defect Predictors" by Burak Turhan, Ayse Bener and Tim Menzies. Profes 2010 2010 . .
48. "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> .
49. 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>.
50. 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 from <http://menzies.us/pdf/09bfc.pdf> .
51. 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,
52. 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> .
53. 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 from <http://menzies.us/pdf/09irrf.pdf> .
54. 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> .

55. 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> .
56. 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> .
57. 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> .
58. 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>.
59. 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>.
60. T. Menzies and A. Marcus. Automated severity assessment of software defect reports. In ICSM'08, 2008. Available from <http://menzies.us/pdf/08severis.pdf>.
61. 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>.
62. 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>.
63. 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 Artificial Intelligence, Robotics and Automation in Space, 2008. Available from <http://menzies.us/pdf/08antares.pdf>.
64. T. Menzies, O. Elrawas, J. Hihn, M. Feathear, B. Boehm, and R. Madachy. The business case for automated software engineering. 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.
65. 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>.
66. 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>.
67. 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>.
68. K. Lum, J. Hihn, and T. Menzies. Studies 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>.
69. 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>.
70. M.S. Fisher and T. Menzies. Learning ivv strategies. In HICSS'06, 2006. Available from <http://menzies.us/pdf/06hicss.pdf>.
71. 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>.
72. 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>.
73. 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>.
74. 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>.
75. D. Owen and T. Menzies. Lurch: a lightweight alternative to model checking. In SEKE '03, 2003. Available from <http://menzies.us/pdf/03lurch.pdf>.
76. 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>.
77. 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>.
78. 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>.
79. 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>.
80. 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>.

81. 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>.
82. 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>.
83. 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>.
84. 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>.
85. 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>.
86. 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>.
87. 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>.
88. 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>.
89. 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>.
90. 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>.
91. 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>.
92. 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>.
93. 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>.
94. T. Menzies and J.D. Kiper. Better reasoning about software engineering activities. In ASE-2001, 2001. Available from <http://menzies.us/pdf/01ase.pdf>.
95. Tim Menzies, Bojan Cukic, Harhsinder Singh, and John Powell. Testing nondeterminate systems. In ISSRE 2000, 2000. Available from <http://menzies.us/pdf/00issre.pdf>.
96. 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>.
97. 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>.
98. 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>.
99. 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.
100. 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>.
101. M. Postema, T.J. Menzies, and X. Wu. A decision support tool for tuning parameters in a machine learning 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>.
102. M. Postema, 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>.
103. S. Ramakrishnan, T. Menzies, M. Hasslinger, P. Bok, H. McCarthy, B. Devakadacham, and D. Moulder. On building an effective measurement system for oo software process, product and resource tracking. In Tools Pacific, 1996, 1996.
104. S. Ramakrishnan, T. Menzies, M. Hasslinger, P. Bok, H. McCarthy, B. Devakadacham, 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>.

105. S. Ramakrishnan and T. Menzies. An ongoing experiment in o-o software process and product measurements. In Proceedings SEEP'96, New Zealand, 1996.
106. 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>.
107. T.J. Menzies. On the practicality of abductive validation. In ECAI '96, 1996. Available from <http://menzies.us/pdf/96ok.pdf>.
108. 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>.
109. 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>.
110. 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.
111. 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>.
112. 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.
113. T.J. Menzies and R. Spurrett. How to Edit "r" or a Black-box Constraint Based Framework for User i; Interaction with Arbitrary Structures. In Tools Pacific 12, pages 213–224. Prentice Hall, 1993. Available from <http://menzies.us/pdf/tools93.pdf>.
114. P. Haynes and T.J. Menzies. C++ is Better than Smalltalk? In Tools Pacific 1993, pages 75–82, 1993.
115. 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>.
116. 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>.
117. 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>.
118. 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.
119. Parametric analysis of a hover test vehicle using advanced test generation and data analysis.
120. 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>.
121. 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>.
122. 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>.
123. 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>.
124. 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>.
125. 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>.
126. 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>.
127. T.J. Menzies. Maintaining procedural knowledge: Ripple-down-functions. In Proceedings of AI '92, Australia, 1992. Available from <http://menzies.us/pdf/ai92.pdf>.
128. 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.
129. T.J. Menzies. Isa object part-of knowledge representation? In Proceedings AI '90, 1990.
130. 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.
131. T.J. Menzies and C. Worral. Worlds in prolog. In Proceedings of AI '87, 1987.
132. 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, Thomas Zimmermann: Software Analytics: What's Next? IEEE Software 35(5): 64-70 (2018)
2. Rafael Prikladnicki, Tim Menzies: From Voice of Evidence to Redirections. IEEE Software 35(1): 11-13 (2018)
3. Olga Baysal, Tim Menzies: Proceedings of the 3rd ACM SIGSOFT International Workshop on Software Analytics, SWAN@ESEC/SIGSOFT FSE 2017, Paderborn, Germany, September 4, 2017. ACM 2017, ISBN 978-1-4503-5157-7
4. Marouane Kessentini, Tim Menzies: A guest editorial: special issue on search based software engineering and data mining. Autom. Softw. Eng. 24(3): 573-574 (2017)
5. Ye Yang, Davide Falessi, Tim Menzies, Jairus Hihn: Actionable Analytics for Software Engineering. IEEE Software 35(1): 51-53 (2018)
6. Tim Menzies, Cross-Project Data for Software Engineering, IEEE Computer December, 2015, p6
7. 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
8. 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
9. R. Harrison, T. Menzies, Guest editorial: special issue on realizing AI synergies in software engineering, Automated Software Engineering, 22(1), 2015
10. Tim Menzies: Beyond Data Mining. IEEE Software 30(3): 92 (2013)
11. Tim Menzies, Thomas Zimmermann: Software Analytics: So What? IEEE Software 30(4): 31-37 (2013)
12. Tim Menzies, Thomas Zimmermann: The Many Faces of Software Analytics. IEEE Software 30(5): 28-29 (2013)
13. 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)
14. Tim Menzies, Martin Shepperd: Special issue on repeatable results in software engineering prediction. Empirical Software Engineering 17(1-2): 1-17 (2012)
15. T. Menzies. 21st century AI: proud, not smug. IEEE Intelligent Systems, 2003. Available from <http://menzies.us/pdf/03aipride.pdf>.
16. 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/99eeked.pdf>.
17. 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>.
18. 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. Zhe Yu, Tim Menzies. Data Balancing for Technologically Assisted Reviews: Undersampling or Reweighting. CLEF (Working Notes) 2017
2. Neil A. Ernst, John Klein, George Mathew, Tim Menzies: Using Stakeholder Preferences to Make Better Architecture Decisions. ICSA Workshops 2017: 133-136
3. 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)
4. Learning the task management space of an aircraft approach model, AAAI 2014 Spring Symposium, Joseph Krall, Tim Menzies, Misty Davis.
5. 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
6. 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)
7. 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 .

8. B. Turhan, A. Bener, and T. Menzies. Nearest neighbor sampling for cross company defect predictors. In Proceedings, DEFECTS 2008, 2008. hW.
9. 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>.
10. 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>.
11. Y. Jiang, B. Cukic, and T. Menzies. Does transformation help? In Defects 2008, 2008. Available from <http://menzies.us/pdf/08transform.pdf>.
12. 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>.
13. 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>.
14. T. Menies, K. Lum, and J. Hihn. The deviance problem in effort estimation. In PROMISE, 2006, 2006. Available from <http://menzies.us/06deviations.pdf>.
15. 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>.
16. 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>.
17. 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>.
18. 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>.
19. T. Menzies, S. Setamanit, and D. Raffo. Data mining from process models. In PROSIM 2004, 2004. Available from <http://menzies.us/pdf/04dmpm.pdf>.
20. 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>.
21. 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>.
22. 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>.
23. 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>.
24. 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>.
25. 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>.
26. 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>.
27. 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>.
28. 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>.
29. 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>.
30. 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>.
31. 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>.

32. 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>.
33. 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>.
34. 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>.
35. 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>.
36. 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>.
37. 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>.
38. 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>.
39. 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>.
40. 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>.
41. 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>.
42. T.J. Menzies. Evaluation issues with critical success metrics. In Banff KA '98 workshop., 1998. Available from <http://menzies.us/pdf/97langevl.pdf>.
43. T.J. Menzies. Evaluation issues for problem visual programming languages, 1998. Banff KA workshop, 1998. Available from <http://menzies.us/pdf/97evalvp.pdf>.
44. T.J. Menzies. Evaluation issues for problem solving methods. In Banff Knowledge Acquisition workshop, 1998, 1998. Available from <http://menzies.us/pdf/97eval.pdf>.
45. 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>.
46. 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>.
47. 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>.
48. 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>.
49. 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>.
50. 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>.
51. 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/96ecaime.pdf>.
52. 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>.
53. 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>.
54. T.J. Menzies and P. Compton. The (extensive) implications of evaluation on the development of knowledge-based systems.
55. In Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge Based Systems., 1995. Available from <http://menzies.us/pdf/banff95.pdf>.

56. 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.
57. 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>.
58. 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>.
59. T.J. Menzies. The complexity of model review. In DX-93: The International Workshop on Principles on Model-Based Diagnosis, 1993.
60. 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.
61. 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 25-27, 1992.
62. 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.
63. 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.
64. 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.
65. T.J. Menzies. Concerning the user of procedural construct as a knowledge acquisition technique. In IJCAI '91 Knowledge Acquisition Workshop, 1991.
66. 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.
67. T. Menzies. Applications of computational intelligence to quantitative software engineering, 2001. Available from <http://menzies.us/pdf/01quase.pdf>.
68. 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>.
69. 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>.
70. 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>.

RESEARCH FUNDING

| CARREER TOTALS | grant: sole PI | co-PI (expended by TM:) | Research expenditure | Grants- Co-PI (total) | Research expenditure |
|-------------------|-------------------|--------------------------------|-------------------------|--------------------------|-------------------------|
| A | B | C=A + B | D | E=C+D | |
| \$4,131,722.00 | \$2,227,070.25 | \$8,522,002.25 | \$3,682,052.00 | \$13,055,582.25 | |

POST-TENURE:

| start | finish | Funding body | Name | gift? | grant: sole PI | co-PI (expended by TM:) | Research expenditure | Grants- Co-PI (total) | Research expenditure |
|-------|--------|-----------------------|--|-------|-------------------|--------------------------------|-------------------------|--------------------------|-------------------------|
| A | B | C=A + B | D | E=C+D | | | | | |
| 2021 | 2021 | Crown Consulting | Model-Based Engineering | | | | 45,800 | | \$45,800 |
| 2020 | 2021 | IBM | Global University Program Academic Award | | | | 40,000 | | \$40,000 |
| 2020 | 2021 | LAS (NSA) | Deeep learning | | | | 106,101 | | \$106,101 |
| 2020 | 2021 | Facebokk | Good via construction (Y2) | | | | 50,000 | | \$50,000 |
| 2019 | 2022 | NSF | Mega-Transfer | | | | 499,371 | | \$499,371 |
| 2019 | 2020 | Lexis Nexis | Leverage (year2) | y | | | 120,000 | | \$120,000 |
| 2019 | 2022 | NSF | Empirical SE for Computational Science | | | | 592,129 | | \$592,129 |
| 2019 | 2022 | NSF | Science of Vulnerability Detection | | | 249,999 | 249,999 | 249,999 | \$499,387 |
| 2017 | 2021 | NSF | Autotuning | | | \$450,000 | \$450,000 | \$450,000 | \$900,000 |
| 2020 | 2020 | LAS (NSA) | fariness is a choce | | | | 255,000 | | \$255,000 |
| 2019 | 2020 | Darpa | sail-on v&v ai | | | | 70,000 | | \$70,000 |
| 2019 | 2020 | NSF | Workforce Empowerment | | | | 98,000 | | \$950,139 |
| 2019 | 2020 | Facebook | Good via construction | y | | | 50,000 | | \$50,000 |
| 2019 | 2019 | LAS (NSA) | How safe is this conclusion | | | | 186,809 | | \$186,809 |
| 2018 | 2018 | Lexis Nexis (Atlanta) | Entiry recognition | y | \$40,000 | | 40,000 | | \$40,000 |
| 2018 | 2018 | IBM | Analytics: SE faculty award | y | \$40,000 | | 40,000 | | \$40,000 |
| 2018 | 2019 | Lexis Nexis (Raleigh) | configure cloud + test cases | y | \$110,000 | | 110,000 | | \$110,000 |
| 2018 | 2019 | NSF | Empirical SE for Computational Science | | \$124,600 | | 124,600 | | \$124,600 |
| 2018 | 2018 | LAS (NSA) | How to make a magican | | \$35,000 | | 35,000 | | \$35,000 |
| 2020 | 2020 | Lexis Nexis | Smoke (text mining to predict M&A) | | | | 50,000 | | \$50,000 |
| 2017 | 2017 | IBM | Automated SE: Faculty grant | y | \$40,000 | | 40,000 | | \$40,000 |
| 2015 | 2018 | Lexis Nexis (Raleigh) | | y | \$120,000 | | 120,000 | | \$120,000 |
| 2017 | 2017 | Lexis Nexis (Atlanta) | Validation lab | y | \$35,000 | | 35,000 | | \$35,000 |
| 2017 | 2017 | Lexis Nexis (Atlanta) | The agreement machine | y | \$35,000 | | 35,000 | | \$35,000 |
| 2017 | 2017 | LAS (NSA) | Privitized data sharing | | \$35,000 | | 35,000 | | \$35,000 |
| 2016 | 2016 | IBM | Automated SE: Faculty grant | y | \$40,000 | \$0 | 40,000 | \$0 | \$40,000 |
| 2016 | 2018 | NSF | Reu: Science of Software | | | \$10,000 | 10,000 | \$345,365 | \$355,365 |
| 2016 | 2016 | Lexis Nexis | Optimization of ML for Big Data | y | \$50,000 | \$0 | 50,000 | \$0 | \$50,000 |
| 2016 | 2016 | SEI | Optimization business process | | \$75,000 | \$0 | 75,000 | \$0 | \$75,000 |
| 2015 | 2017 | NCDSA | Share Care Beware | | \$60,000 | \$0 | 60,000 | \$0 | \$60,000 |
| 2015 | 2018 | Lexis Nexis | Validation lab | y | \$120,000 | | 120,000 | | \$120,000 |
| 2015 | 2015 | JPL | Effort Estimation (year2) | | \$30,000 | | 30,000 | | \$30,000 |
| 2014 | 2015 | Lexis Nexis | Scripting for Big data | y | \$50,000 | | 50,000 | | \$50,000 |
| 2013 | 2017 | NSF | Transfer Learning in SE | | | 622,030 | 622,030 | 529,773 | \$1,151,803 |
| 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 since tenure (2006+) | | \$2,281,761.00 | \$2,227,070.25 | \$6,672,041.25 | \$3,682,052.00 | \$11,205,621.25 |

PRE-TENURE

| | | | | | | | | | |
|----------------------|------|-------------------|--|--|----------------|--------|----------------|--------|----------------|
| 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? | | \$126,161 | | \$126,161 | | \$126,161 |
| 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! | | \$47,800 | | \$47,800 | | \$47,800 |
| 2002 | 2003 | NASA | A spectrum of IV&V techniques | | \$200,000 | | \$200,000 | | \$200,000 |
| 2002 | 2002 | NASA | Better risk modelling | | \$29,000 | | \$29,000 | | \$29,000 |
| 2001 | 2001 | NASA | Tree query languages | | \$29,000 | | \$29,000 | | \$29,000 |
| 2000 | 2000 | Canada Res. Coun. | NSERC grant | | \$83,000 | | \$83,000 | | \$83,000 |
| 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) | | | | | \$1,849,961.00 | \$0.00 | \$1,849,961.00 | \$0.00 | \$1,849,961.00 |

| Project ID | Title | Investigator(s) | PI Department | Sponsor | Status | Amount Awarded/ Contract Number | Current Award Period |
|---|--|--|---------------------------|---|-----------|---|-------------------------------|
| (1419) 2021-1379 Details | LAS DO2 Menzies- MLI (Confidential) | Menzies, Timothy James | Computer Science (141901) | Laboratory for Analytic Sciences | Awarded | \$53,133.00 H98230-19-D-0012 (52.0% - B -) | 12/23/2020 through 12/31/2021 |
| (1419) 2021-0833 Details | SaTC: CORE: Small: Defending Malicious Software Detectors via MASS CONFUSION | Menzies, Timothy James | Computer Science (141901) | National Science Foundation (NSF) | Proposed | | |
| (1419) 2021-0279 Details Ewendt, Maks | Model-Based Systems Engineering with Machine Learning for Test Case Generation | Menzies, Timothy James | Computer Science (141901) | Crown Consulting (Prime--US Dept. of Defense (DOD)) | Proposed | | |
| (1419) 2021-0238 Details Ewendt, Maks | Teaming Agreement with Crown Consulting | Menzies, Timothy James | Computer Science (141901) | Crown Consulting (Prime--US Dept. of Defense (DOD)) | File Only | \$0.00 | 08/06/2020 through 08/05/2021 |
| (1419) 2021-0119 Details Wallace, Virginia | SMOKE (anyone can see the fire, but when can you notice the smoke?) | Menzies, Timothy James | Computer Science (141901) | LexisNexis | Awarded | \$47,082.00 1 (52.0% - B -) | 05/01/2020 through 12/31/2020 |
| (1602) 2020-2757 Details | B1: WE-RISE: Workforce Empowerment and Retraining through Internet of Solutions for Employment | Ding, Huiling Chi, Min Fang, Xiaolei Menzies, Timothy James Shen, Xipeng | English (160201) | National Science Foundation (NSF) | Proposed | | |
| (1419) 2020-2601 Details | CHAIIn | Menzies, Timothy James | Computer Science (141901) | Quantum Ventura Inc. (Prime--Defense Advanced Research Projects Agency (DARPA)) | Proposed | | |
| (1419) 2020-0238 Details | LR2: Ultra-fast Novelty Recognition and Repair for Deep Learning | Menzies, Timothy James | Computer Science (141901) | Quantum Ventura Inc. (Prime--Defense Advanced Research Projects Agency (DARPA)) | Awarded | \$72,675.00 2020-0238 (52.0% - B -) | 01/30/2020 through 05/28/2020 |

| Internal Funding | | | | | | |
|-----------------------------------|--|------------------------|------------------|----------------------------------|-----------|-------------------------------|
| 2017-1422 | LAS DO7 Menzies - 3.4 Cybersecurity Second Order Effects | Menzies, Timothy James | Computer Science | Laboratory for Analytic Sciences | \$186,809 | 01/01/2017 through 12/31/2018 |
| 2019-1060 | LAS DO1 Menzies - 2.4 Analytics, AI and Machine Learning | Menzies, Timothy James | Computer Science | Laboratory for Analytic Sciences | \$181,110 | 01/17/2019 through 12/31/2020 |
| Total internal funding: \$367,919 | | | | | | |

| Pending Proposals (including pre-proposals) | | | | | |
|---|---|---|----------------------|--|-----------|
| 2019-2049 | Artificial Intelligence Fighting Against Cyberattacks: Instant Prevention of Software Vulnerabilities | Menzies, Timothy James | Computer Science C&G | University of Wollongong (Prime - Australian Research Council) | \$0 |
| 2019-2780 | FAI: Fairness is a Choice (and Not Choosing is Unfair) | Menzies, Timothy James | Computer Science | National Science Foundation (NSF) | \$705,216 |
| 2020-0074 | Ethics and Safety of Systems of Autonomous, Intelligent Agents in Society | Singh, Munindar P. Roberts, David L. Doyle, Jon Wilson, Mark A. Struett, Michael Mayer, Roger Dutta, Rudra Desmarais, Sarah Dubljevic, Veljko Bauer, William A. Menzies, Timothy James Bardaka, Eleni Nam, Chang S Chen, Crystal Gillan, Douglas J Ding, Huiling Williams, Laurie A. Das, Anupam Bozkurt, Alper Yusuf Barbee, Lindsey McGowen List, George F. | Computer Science | National Science Foundation (NSF) | \$2 |
| 2020-0238 | LR2: Ultra-fast Novelty Recognition and Repair for Deep Learning | Menzies, Timothy James | Computer Science | Quantum Ventura Inc. (Prime - Defense Advanced Research Projects Agency (DARPA)) | \$72,675 |
| Total of pending proposals: \$777,893 | | | | | |

| 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 |
| 2016-0702 | SHF:Medium:Holistic Scalable Autotuning for Software Engineer Data Analytics | Menzies, Timothy James Shen, Xipeng | Computer Science | National Science Foundation (NSF) | \$1,200,000 |
| 2016-0738 | 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 |
| 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 |
| 2017-1061 | SHF: MEDIUM: Is There Wisdom in the (Qualified) Crowd? | Stolee, Kathryn Thomasset Menzies, Timothy James | Computer Science | National Science Foundation (NSF) | \$1,199,261 |
| 2017-1445 | SaTC: Core: Small: Are Vulnerability Prediction Models Possible for Practical Use? | Williams, Laurie A. Menzies, Timothy James | Computer Science | National Science Foundation (NSF) | \$499,486 |
| 2018-0729 | SHF: Medium: Better Software Analytics by Combining Human and Artificial Intelligence | Menzies, Timothy James | Computer Science | National Science Foundation (NSF) | \$905,037 |
| 2018-1293 | SHF: Small: Jumpstarting Next Generation Vulnerability Prediction Models | Menzies, Timothy James Williams, Laurie A. | Computer Science | National Science Foundation (NSF) | \$499,857 |
| 2019-2050 | CCRI:Medium:Collaborative Research:SAInT: Bridging the Gap from Research to Practical Advice | Menzies, Timothy James | Computer Science | National Science Foundation (NSF) | \$718,159 |
| Total of non-funded proposals: \$7,983,466 | | | | | |
| Miscellaneous Activities | | | | | |
| Total miscellaneous activity funding: \$0 | | | | | |

IV. EXTENSION AND ENGAGEMENT WITH CONSTITUENCIES OUTSIDE THE UNIVERSITY

- Crown Consulting, STTR phase 1
- Lexis Nexis, commercial text mining
- Lexis Nexis: test case prioritization , Raleigh campus
- Lexis Nexis: cloud configuration, Raleigh campus
- Lexis Nexis: test case prioritization, Raleigh campus
- IBM, data mining work with research triangle. Faculty award 2017.
- IBM, data mining work with research triangle. Faculty award 2016.
- Lexis Nexis, text mining work, 2015, 2016, 2017
- Jet Propulsion Lab, effort estimation research. 2002 to present. Funded research 2007, 20014 to 2015
- 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

TECHNOLOGICAL AND MANAGERIAL INNOVATION

A. TECHNOLOGY TRANSFER

Workshop Organizer/Facilitator:

- RAISE pc-chair 2019
- ROSE festival organizer (DSE'18, ICSE'19)
- FSE'18 artifacts chair
- SSBSE co-PC chair, 2017
- FSE SWAN workshop, co-chair, 2017
- Artifacts track, FSE'16
- Artifacts track, ICMSE, 16
- 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

Here is the text of my 2018 IEEE Fellow application (which was successful).

Internationally known for revolutionary advances exploring the synergy between artificial intelligence (AI) and software engineering (SE), Dr. Menzies has authored four books and over 260 refereed publications. His publications, with over 9000 citations, have appeared in leading journals and proceedings of prestigious conferences. He has supervised seven students earning PhDs and 23 MS thesis students. Dr. Menzies' distinctive contributions have had enormous impact for SE researchers and practitioners in software quality prediction and software optimization.

SOFTWARE QUALITY PREDICTION

Because software plays a critical role in industry, government, and society itself, improving software quality is critical. In landmark papers in 2006 and 2007, Dr. Menzies was an early pioneer in applying data mining and AI to software quality predictors, introducing a method which identified software modules likely to contain defects. This method had a 71 percent mean probability of defect detection, significantly higher than the code inspections commonly used in software practice.

In his software quality prediction research, Dr. Menzies identified a serious problem: often, the analysis in SE papers is not reproducible because data underlying the analysis is unavailable. To address this problem, Dr. Menzies developed PROMISE, a public data repository of software data, in 2005 publishing a paper introducing PROMISE and co-founding the PROMISE workshop, so successful it became a conference in 2008. Today, the PROMISE repository contains hundreds of data sets used in thousands of papers by researchers around the world.

OPTIMIZATION OF SOFTWARE-INTENSIVE SYSTEMS

Dr. Menzies is a pioneer in applying data miners to optimize software-intensive systems. In 2002, he discovered that analyzing such systems with data miners augmented with genetic algorithms led to faster analysis and better optimizations. Even for systems with millions of configuration options, Dr. Menzies' optimizers quickly learn how to make code run quicker, make

web servers handle more traffic, and compile programs faster. Dr. Menzies' optimizers have been applied at NASA for reasoning about safety-critical aerospace software.

Dr. Menzies has also applied his optimization techniques to understand the unstructured textual components of software artifacts and software research papers. His was one of the earliest successful efforts applying text mining and AI to the notes of software test engineers. By identifying anomalous reports that required a second opinion, he could increase assurance of NASA systems while reducing the overall effort required to achieve that assurance. Recently, he has designed tools that can review 10,000s of papers to learn the structure of the SE scientific community. These tools can guide researchers and practitioners to find relevant work that might otherwise be overlooked.

Dr. Menzies' contributions to SE and AI are widely recognized. For his research, in 2017, Dr. Menzies received the MSR (Mining Software Repositories) Foundational Contribution Award as "Recognition of fundamental contributions in the field of data mining software repositories which helped others advance the state of the art." International databases of scholarly achievement rank Dr. Menzies number three world- wide both in software analytics and in SE and data mining. Recently, Dr. Menzies clustered 35,000 papers from the last 25 years of top-SE journals and conferences. In the "software metrics" cluster, Dr. Menzies is the top-ranked author. In the papers from top-ranked venues, Dr. Menzies' h-index of 48 places him number 11 overall.

Dr. Menzies' contributions have had world-wide impact in software practice. In 2005, Turkish researchers found that when commercial teams restricted code inspections to 25 percent of the files identified by Dr. Menzies' methods, they detected 88 percent of the existing code defects. In 2005, his students commercialized his defect detection methods in the Predictive tool suite, subsequently purchased by companies such as Chevron, Northrop Grumman, LogLogic, Inc., and Compagnie Financière Alcatel, to find code defects. In 2017, the US Software Engineering Institute used Dr. Menzies' optimizers to guide discussions about costly updates to Department of Defense software.

NASA has benefited enormously from Dr. Menzies' research. In 2005, as science chair at a NASA facility, he received a commendation award from NASA's Chief of Mission Assurance saying: "...A great researcher in his own right, ...Tim has raised the bar on quality and level of work [expected] from our researchers." NASA used his algorithms in 2008 to find violations in Space Shuttle launch requirements; in 2010, to quickly explore the design of next-generation new Air Traffic Management concepts; and in 2017 to find better monitoring strategies for pilots flying planes in safety-critical situations. In 2016, based on Dr. Menzies' research, NASA's Jet Propulsion Laboratory created the NASA Analogy Software Cost Model as its official tool for predicting software development costs.

Evidence of Technical Accomplishment

- Tim Menzies, Jeremy Greenwald, Art Frank, "Data mining static code attributes to learn defect predictors," IEEE Transactions on Software Engineering, Vol. 33 (1), 2-13, 2007. Dr. Menzies is a pioneer in the development of predictors of software quality learned from data miners. A notable finding of this paper is that Dr. Menzies' methods (including decision trees and Bayesian learning) have a 71 percent mean probability of defect detection--a rate significantly higher than human manual inspections. The paper, with more than 920 Google Scholar citations, is one of the 100 most cited papers in software engineering. Moreover, nine of the 50 most cited papers in the IEEE Transactions on SE (2012-2017) use methods and/or data from the databases used by this paper. Methods for software defect prediction introduced in the paper have been applied commercially around the world. In this paper, Dr. Menzies was the lead researcher—he defined the problem, the technical approach, and designed and coded all of the experiments.
- Martin S. Feather and Tim Menzies, "Converging on the optimal attainment of requirements," Proceedings, IEEE Joint International Conference on Requirements Engineering, 2002. Although optimization methods for numerical systems have been used widely, applying these methods is often ineffective in complex software systems where each "if" statement divides the software into regions with different properties. For software, Dr. Menzies found that applying non-numeric optimizers, e.g., simulated annealing or genetic algorithms, is effective. This paper is the first of its kind to reason about solutions to software requirement problems on the Pareto frontier. As witnessed by many papers in the last two years, this method is now widely used by researchers in the software requirements community. For this paper, Dr. Menzies led the AI-part of the research, and designed and implemented the AI algorithm used in the analysis.
- Tim Menzies, Andrian Marcus, "Automated severity assessment of software defect reports," IEEE International Conference on Software Maintenance," 2008. This paper, with over 185 citations, describes one of the earliest successful efforts applying text mining methods to the notes of software test engineers. The method introduced in the paper identifies anomalous reports requiring a second opinion, thus increasing software quality assurance while

reducing the overall effort required to achieve that assurance. For this work, Dr. Menzies was the lead researcher, defining the overall vision of the paper, as well as building the tools and running all of the experiments.

Natural Language Understanding

- Zhe Yu, Nicholas Kraft, Tim Menzies "Finding Better Active Learners for Faster Literature Reviews". Empirical SE Journal, to appear 2018. This is the first SE application using incremental text mining methods to learn what a reader wants to read. Dr. Menzies showed that a) supposed state-of-the-art text miners from other domains perform poorly for SE, and b) a new method called FASTREAD can quickly guide researchers and practitioners to relevant work that might otherwise be overlooked. Achieving Generalizability in Software Engineering Research
- Burak Turhan, Tim Menzies, Ayse Basar Bener, Justin S. Di Stefano, "On the relative value of cross-company and within-company data for defect prediction," Empirical SE, vol. 38(6), 1403-1416, 2012. This paper shows that useful models for a project can be built by carefully selecting the most relevant examples from other projects. The paper, with more than 180 citations, is one of the five most cited articles of all time in the Empirical SE journal.

Adjusting Learners to Human Needs

- Abdel Salam Sayyad, Tim Menzies, and Hany Ammar, "On the value of user preferences in search-based software engineering: A case study in software product lines", International Conference on Software Engineering, 2013. This paper (134 citations) shows that while most, but not all, optimizers used in software engineering are highly insensitive to complex sets of user preferences, for complex requirements problems, goal-aware reasoning can achieve much better results than standard optimizers used in software engineering.
- Abdel Salam Sayyad, Joseph Ingram, Tim Menzies, Hany Ammar, "Scalable product line configuration: A straw to break the camel's back". Automated Software Engineering Conference, 2013. This paper (86 citations.), which extends the previous paper, is one of the five most cited papers in the IEEE Automated Software Engineering Conference in the last five years. By exploiting the richness of human preferences, the method introduced in this paper can extract usable designs from a space of thousands of goals and hundreds of thousands of constraints. Learning Using Many Opinions
- E Kocaguneli, Tim Menzies, JW Keung, "On the value of ensemble effort estimation," IEEE Transactions on Software Engineering, Vol. 38(6), 1403-1416, 2012. Even though ensemble techniques are widely applied in other domains, they are rarely used in SE. This paper (140 citations) showed that any single predictor was less trustworthy than using twelve elite models implemented from an ensemble of 90 learners.

Uncovering Errors in Data Mining

- Tim Menzies, Alex Dekhtyar, Justin Di Stefano, Jeremy Greenwald, "Problems with precision" IEEE Trans SE, 2007. In this paper (155 citations), Dr. Menzies describes a previously undocumented, subtle, and dangerous aspect of precision in a widely-used performance measure. Curiously, this problem had not been previously reported despite the measure's widespread use.

Better Optimizers Using Data Miners

- Tim Menzies, Zach Milton, Burak Turhan, Bojan Cukic, Yue Jiang, Ayse Basar Bener, "Defect prediction from static code features: Current results, limitations, new approaches". Automated Software Engineering, 2010. This paper (197 citations) introduces "WHICH" a meta-learner framework that can be quickly customized for different business goals. Measured in terms of specific user goals, WHICH performs better than many standard learners. Human-Understandable Data Mining Results
- Tim Menzies, Ying Hu, "Data mining for very busy people", IEEE Computer, Vol. 36(11), 2003. Cognitive scientists and researchers studying human decision-making note that humans often use simple models rather than complex ones. This paper (132 citations) describes Dr. Menzies' TAR2 data miner which generates tiny human-readable models, useful for describing to humans many seemingly complex software engineering problems.

SERVICE TO THE UNIVERSITY AND PROFESSIONAL SOCIETIES

A. UNIVERSITY SERVICE

- Chair, Carla Savage awards committee
- Chair, Search committee, SE faculty, 2019
- Member, CSC Graduate Program Oversight Committee. 2016-present
- Member, Strategic Planning Committee, 2016-
- Member search committees, CSC, 2016
- Worked on the graduate recruiting weekend, March 2017.
- Member, University faculty scholars review committee (Sept 8, 2017).
- Volunteer, Open Day, March 2015
- Speaker, Graduate research seminar series (CS), November '14

B. NATIONAL AND INTERNATIONAL SERVICE

- Editor in chief
 - Automated Software Engineering journal
- Editorial Board
 - IE Software journal
 - Journal of Software Systems, 2016-
 - 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)
 - Transactions Software Methodologies
- Associate Editor
 - IEEE Transactions on Software Engineering, 2011-2017
- General Chair
 - IEEE International Conference Software Maintenance and Evolution, 2016
- Program Chair/Co-Chair:
 - RAISE 2019 PC-chair
 - **PROMISE 2020 PC-chair**
 - ROSE organizer ICSE'19 FSE'18
 - 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:
 - (2018) Automated Software Engineering special issue best papers ASE2017
 - (2017) IEEE Software, special issue on Actionable Analytics
 - (2017) Empirical Software Engineering journal, special issue, Big Data and SE
 - (2017) Automated software Engineering journal, Special issue on Next Generation Search-based SE
 - (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:
 - ICSE Artifacts chair, 2020
 - Artifacts chair, FSE'18
 - Data challenge, RE'18
 - PC-chair: SSBSE 2017,
 - Co-PC chair FSE SWAN 2017
 - Artifacts chair: FSE 2016
 - Artifacts chair: ICSME 2016
- Program Committee:
 - 2021:
 - ICSE'21
 - IJCAI'21
 - AAAI'21
 - 2019:
 - IEEE Fellow award committee (for 2020).
 - ICSE,18, Msr award committees 2019,
 - 2018:
 - ICSE,18, Msr award committees 2018, SSBSE'18, ESEM 2018
 - 2017:
 - SoftwareMining'17, ICSE'17 (artifacts), PROMISE'18, MSR'18, ESEM'18, SPLC'17, EASE'17
 - 2016:
 - ASE'16, BIGDSE'16, EASE 2016, ESEM2016, ICSE-SRC 2016, ISSRE 2016, PROMISE 2016, RAISE 2016, SSBSE'16, SCORE 2016
 - 2015:
 - Ase'15, BigDSE'15, Ease'15, EsPreSSE'15, Esem'15, Fse'15, Gecco'15, Icp'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 , IEEE Transactions on Evolutionary Computation