

---

**Dossier**  
**Tim Menzies**  
**2018-2019**

Department of Computer Science  
North Carolina State University  
email: [tjmenzie@csc.ncsu.edu](mailto:tjmenzie@csc.ncsu.edu);  
URL: <http://menzies.us>

---

## STATEMENT OF MUTAL EXPECTATION

Tim Menzies, Computer Science, NC State

### **I. Introduction**

#### **A. Statement of Mutual Expectations (SME)**

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

*Faculty input:* Expectations are standard with respect to 5th year Full Professors.

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

#### **ii) Teaching (academic activities) – Teaching Responsibilities**

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

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

I will be teaching the graduate software engineering (CSC 510), under-graduate programming languages, and a special topic topics course: automated software engineering (a 500-level, 700-level). I will officially advise at least graduate students. I will be serving on several exam and dissertation committees.

#### **iii) Scholarship (research and innovation) – Research Areas**

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

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

Research areas include software engineering and data science and automated software engineering.

#### **iv) Professional Activities (service)**

---

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

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

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

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

- Serve on the program committee for several major conferences and workshops, most notably ICSE'20, ASE'19, FSE'19, EMSE'19,
- Continue to review papers from numerous top-level conferences.
- I will write white papers (seeking industrial funding) and 1-2 NSF grant proposals.

A handwritten signature in blue ink that reads "Tim Menzies". The signature is written in a cursive, slightly stylized font.

---

Prof. Tim Menzies

## **BRIEF RESUME**

### 1. Education background:

- Ph.D., CS, University of New South Wales, 1995 *Generalized Testing of Knowledge Bases*; Advisor Paul Compton
- Masters of Cognitive Science, University of New South Wales, Australia, 1988
- B.S. Computer Science, University of New South Wales, 1985.

### 2. Professional experience:

- August 2014 to present: Professor, CS, North Carolina State University, Raleigh, NC
- May 2012 to August 2014: Professor, West Virginia University, Morgantown, WV
- February 2006 to April 2012, Associate Professor, West Virginia University, Morgantown, WV
- December 2001 to December 2003, SE research chair, NASA IV&V Facility, West Virginia
- July 2000 to January 2001: Assistant professor, University of British Columbia, Vancouver, CA

### 3. Scholarly and creative activities:

<i>Books</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Authored books	0	0	0	
Edited books and Proceedings	4	4	0	
Refereed book chapters	13	3	0	
<i>Papers, Articles, Patents, Reports, etc.</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	<i>Submitted</i>
Refereed journal articles	78	57	6	7
Refereed magazine articles	18	9	2	
Other magazine articles	-	-	-	
Refereed conference papers	124	63	10	1
Refereed workshop papers	70	15	0	
Refereed panel papers	-	-	-	
Refereed posters/fast abstract	-	-	-	
Technical reports	4	-	-	
Refereed tutorials	4	4	-	
Course pack (with ISBN)	-	-	-	-
News interviews	4	2	-	-
<i>Talks, Presentations</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Keynotes and distinguished speaker	8	8	2	
Other invited talks	10	10	3	
<i>Funded Research, Development and Teaching</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Contracts and Grants	\$9,590,846	\$7,750,885	\$308,628	
Gifts (cash)	\$680,000	\$680,000	\$190,000	
Gifts (in kind)	-	-	-	
Other: PhD Fellowships	-	-	-	
<i>Mentoring and Supervision (see CV for details)</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
PhD (chair/co-chair), graduated	9	8	2	
PhD (chair/co-chair), current	-	12	12	
MS (chair/co-chair), graduated	28	21		
MS (chair/co-chair), current	-	-		
Undergraduate advisees, graduated	7	5		
Faculty mentored	9	9	2	
<i>Courses taught</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Regular undergraduate (3 credits, $10 < x < 100$ students)	9	2	1	
Large undergraduate (3 credits, $x > 100$ students)	5	-	-	
Regular graduate (3 credits, $10 < x < 100$ students)	36	25	2	
<i>Courses created and/or revised in a significant way</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Undergraduate	5	3	1	
Graduate	8	5	3	
<i>Other</i>	<i>Career</i>	<i>Post Tenure</i>	<i>Current Year</i>	
Development of Software Packages	5	2	10	
Creation/Direction of Dept. Facilities – Labs & Centers	5	3	2	
Major awards and recognitions	10	9	2	
Major off-campus services	15	10	4	

4. 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.**

5. Scholarly and professional honors:

1. **ACM TOSEM journal distinguished reviewer 2011-2018 (only person to receive that award for all those years)**
2. **Journal of Software & Systems : best reviewers for 2018 (only 12 such people selected).**
3. Inaugural Mining Software Repositories Foundational Contribution Award., 2017
4. Carol Miller Graduate Lecturer Award, Association for Computing Machinery/Association of Information Technology Professionals, 2016
5. Distinguished reviewer, ACM Transactions on Software Engineering Methodologies, 2016
6. Outstanding reviewer award, journal of Information and Software Technology, 2016
7. Service award from Big Data community: Lexis Nexis, 2015
8. Distinguished reviewer, ACM Transactions on SE Methodologies, 2015
9. WVU College of Engineering, Outstanding Researcher, 2010
10. NASA Commendation for Chief of Mission Assurance, 2004

6. Professional service on campus:

- **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)
- BWVU, computer science, Promotion & Tenure committee (2010-2014)
- WVU, Member, Faculty Search Committees (2010-2013)
- Director, National Archives/WVU project (2009-2011)
- Director, WVU/NASA Research Collaboration (2002-2009)

7. Professional service off campus (see CV for complete list):

- **Artifacts co-chair, ICSE'20**
- **ROSE FESTIVAL ORAGANIZER, FSE'18, ICSE'19**
- **Co-PC 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: SSBSE'17, ICSE NIER'15, ASE'12.
- Editorial board: ACM Transactions on Software Engineering (
- Editorial Board: 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: **ICSE20, MSR/19, FSE'19, ICSE'20 ICSE,18, IEEE Fellow award committee (2020), Msr award commitee 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, Ipcp'15, Issre'15, Msr'15, NasBase'15, Promise'15, Raise'15, Ssbse'15; Previously: MSR'14, ICSE14-demos, ICSE14-mainConference, DAPSE'14, EASE'14, GTSE'14, SAM 2014, SEAA 2014, MSR (2011-2014). ASE (2002-20114), ESEM (2011-2013) • SAM2103, DAPSE'13, ICSE'13: demos ,ASE-Tools'13 , ISSRE'13, GTSE'13, MALIR'13 , Software Mining -2012, 2013 , ISSRE'09, ISSRE'10 and many more dating back to 1991.

## II. TEACHING AND MENTORING OF UNDERGRADUATE AND GRADUATE STUDENTS

### A. TEACHING EFFECTIVENESS

Course	When	Responses/Enrolled	Explanation	Effectiveness	Knowledge	Excellence
CSC 510 (softeng)	Spring 2018	35/68	3.6/4*	3.8/4.1*	3.8/4.3*	3.6/4.1*
CSC 591 023 (foundations software science)	Fall 2018	14/23	42 ^	4.1^	4.2^	4.3^
CSC 485 00 (programming languages)	Spring 2018	15/43	3.0/41.*	3.5/4.2*	3.5/4.3*	2.9/4.0*
CSC-510	Spring 2107	21 / 33	4.1 / 3.9*	4.1 / 4.2*	4.0 / 4.2*	4.1 / 4.0*
CSC-591 (023)	Fall 2017	13 / 25	4.2 / 3.9*	4.3. / 4.2*	4.5 / 4.2*	4.2 / 4.0*
CSC 591-006 (software science)	Fall 2016	8/12	4.8 / 4.1	4.4 / 4.2 *	4.6 / 4.3*	4.8 / 4.1*
CSC591-007 (automated SE)	Fall 2016	36 / 38	4.1 / 4.1*	4.2 / 4.2*	4.3 / 4.3*	4.2 / 4.1*
CSC 510 (SE)	Spring 2016	40 / 57	4.3 / 4.1*	4.4 / 4.2*	4.4 / 4.3*	4.4 / 4.0*
CSC 791/791 (automated SE)	Fall 2015	44 / 52	4.3 / 4.2*	4.4 / 4.3*	4.5 / 4.4*	4.4 / 4.2*
CSC 510 (SE)	Spring 2015	25 / 32	4.2 / 4.2*	4.4 / 4.3*	4.3 / 4.4*	4.3 / 4.2*
CSC 791 (automated SE)	Fall 2014	5 / 5	5.0 / 4.0*	5.0 / 4.0*	5.0 / 4.3*	5.0 / 4.0*

\* departmental average

^ departmental average not reported in class eval report

Student comments:

- **CSCXXX (Fall 2018) :**

- **Dr. Menzies's class is very interesting. Overall, fantastic pace of the course and materials covered. ... Overall, a fantastic course. ... A fantastic instructor. ... Dr Menzies is the most engaging lecturer I have had to date. He brings levity in his lectures. ... Formed his lessons around easy to understand narratives and intuition. .... I feel that the course does a good job of instilling a healthy, scientific**

scepticism when evaluating AI, and imparts tools with which to evaluate AI. ... My suggestion would be to add a homework where folks set up a small experiment to learn how to build a pipeline that has validation and produces evaluation metrics.

- **CSC 495 (Spring 2018):**

- (Notes from Dr Menzies: The scores for this subject were low. Reason: this was the first ugrad subject I had taught since 2014 and I just got too ambitious. I'm teaching the subject in Spring 2019 and this time I am doing much smaller, much more defined, assignments).
- He loves the course, is enthusiastic about it, and knows a ton about it, but he needs to slow down and collect the amazing thoughts in his head because he has trouble conveying them clearly. ... I learned a lot about a variety of different programming languages which has broadened my view of Computer Science in general. The set up of the course itself was disorganized. I think this is partly the fault of the professor for not setting up a comprehensive syllabus, but also because this was a brand new class this semester. ... He is enthusiastic about teaching, but I feel he didn't achieve the goals of the course to me. ... Tim is incredibly enthusiastic about the course, but he can't coherently assemble his thoughts to get out a good lecture. It's so obvious that he knows so much and has so many great ideas, but he just rambles for most of the lecture. If he really organized one topic per lecture and created a coherent "story-line", it'd be soooo much better. ... The materials were similar to his lectures - a non-focused stream of thoughts and examples. Smaller and more focused examples would have made a huge difference. Doing a github for the course was wonderful, but the materials inside could've been cleaned up a lot. ... Most of the examples are so abstract and Dr. Menzies jumps from one thing to another so quickly it can be very easy to get lost. ... I think the due dates should have been more rigid so we don't have assignments due on exam week. ... the project assignment itself was also lacking in detail. This made it very difficult to provide the type of result that the professor was looking for.

- **CSC 510 (Spring 2018):**

- It was a fun, intuitive class. Did not expect SE to be this fun. .... Good course structure, having two projects really enforce the practical aspect of theory explained in the course ... The course has two projects which are intensive and call for the application of the concepts studied in the course. The first project is the one which we developed from scratch. And the second project is the one which is developed by some other team but is made feature-rich or modified positively by another. It's a great concept. This way students get an opportunity of learning how other people have implemented a certain thing and how they add on to it. ... Strength: shares industrial experiences which is very helpful ... It was fun attending his lectures. ... The instructor was pretty enthusiastic and taught with a lot of energy. The lectures were interesting and exciting. ... The professor explains a lot about the related things happening in the world... SE course is too broad. Prof Menzies tries his best to teach the class the way of industry. ... Strengths - "Very knowledgeable" and he knows it. Weaknesses - Very knowledgeable and "he knows it".

- **CSC591-023 (2017):**

- So much enthusiasm OMG!! That is what students need to see in a professor. If it would not have been for his enthusiasm we would not have done our project so well. It just feels good to be in his class. Perfect presentation skills! Glad I took the course.... One of the rare and best course at NC State for data science. Everyone must take it!!! Make it compulsory of Data Science track.... This course was amazing, I learned many things from the lectures and the core concept of the subject was very thorough and with so many hands on experiments we learned the subject very well.... 'This course had everything in perfect order, the core basics, experiment with real life examples, latest most advanced topics to oldest most reliable ones. With the lectures on different papers I learned what are different research topics that is being performed in the industry.' 'This course was very thorough and well planned. The lectures were really enjoyable. And the homeworks were really good and we can learn a lot from them.... 'Very passionate about the material he teaches.'...'Prof. Menzies's classes are very interesting and engaging'

- **CSC510 (2017):**

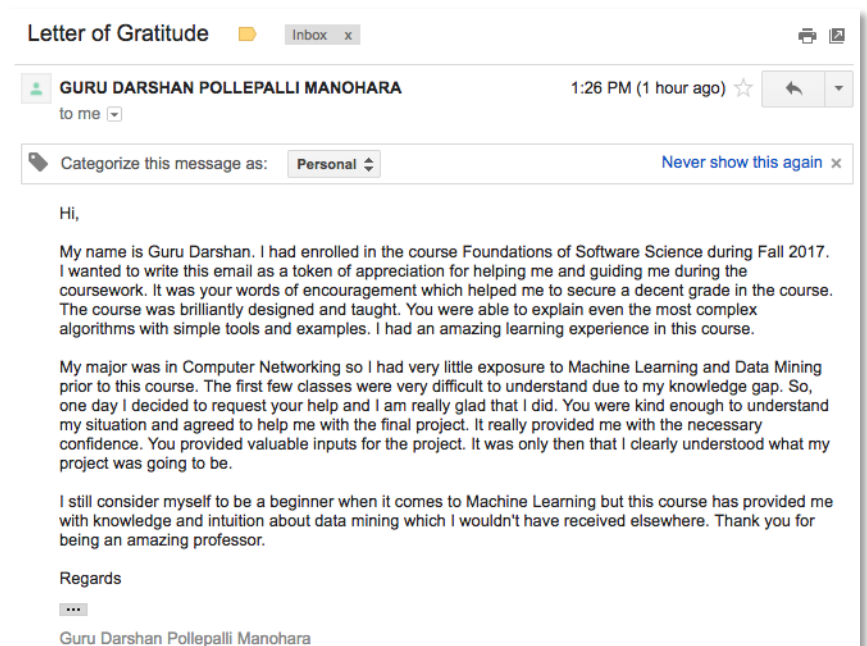
- Treats students with respect, very enthusiastic about teaching the course. ... The instructor is so knowledgeable. He can extend the talk to a lot of things related. Haven't seen any other professor so enthusiastic about the subject. Sometimes I feel hard to follow. ... immense knowledge about the subject. He is very enthusiastic; creates a light hearted mood in the classroom and encourages student participation. ... don't know any other instructor, younger or older, who is as enthusiastic, optimistic and excited about their course. Helps to update oneself to how software is being designed, coded and tested in outside work. Needs, more concentration on coding/projects (including designing and testing/mockup) and how to tackle the problems (faced during software development cycle) - rather than theory. ... At times, there is too much theory. ... I thought the focus on



research didn't really make sense for this course. .... but often he will go off on a tangent, pulling up dense (in terms of content and newness to students) research papers of his own and talk about them at great length.

- Dec 2017, I received the “Letter of Gratitude” show at right from one of my graduate subject’s attendees.
- Prof. Warwick Arden (Provost) mailed me on Mark 16 2016 saying that some students had used the “Thank a Professor” website to comment favorable on my teaching:

- He writes “You efforts with these students area reflection of your dedication to teaching and learning. I congratulate you on this recognition and offer by sincere appreciate for your work with students at NCSU.
- Student comments included “I compare you to an owl because you have a deep connection with wisdom and intuitive knowledge. You have been kind and patient with me and your students. Thank you for everything.”



- **CSC591-006 (2016):**
  - Great teaching! ... Great prof! ... Course taught well. The best lecturer I've had at NC State with Samitova as a close second.... Understood the material well enough to talk off the cuff which means it stayed interesting and well paced. Good interaction with students, always get quick and useful feedback.... Good training of data science and programming.... Knowledgeable and passionate in teaching.... Tthe course laid a very good foundation of data science and how practice it as software scientists, the reading assignments (although not easy) taught me how to analyze a research paper... The course was excellent.
- **CSC591-007 (2016):**
  - Amazing course structure. Some of the concepts were used in my job interviews... What an enthusiastic professor. So funny and makes the class so interesting... Good homeworks and projects. Good materials.... He is amazing and has a lot of knowledge. He is passionate about what he teaches and that's why it makes the class interesting as well. Learnt a lot from him.... He is really passionate about the things he teaches and is a very congenial teacher... Very passionate about the subject which made me excited to learn ASE... It was an amazing course. What I studied in this course, I could not have learned otherwise. I have not seen another professor as enthusiastic as Dr. Menzeis in teaching a class. Plus, he never lets you feel bored in class. Each and every concept I learned were new for me; which made this class different from other classes and challenging as well. ... Very enthusiastic about the material and I loved having the whole course on Github. More instructors should do this. Workshops were also very beneficial more of those please... Professor is very passionate about the subject and is very keen to share what this field has to offer for computer scientists. ... : Almost all the concepts learned in this course were new to me and I guess most other students. This is great knowledge that can be applied in any domain of computer science. Professor and TAs are very helpful with doubts. ...
- **CSC510 (2016):**
  - Awesome professor. ...Nice and kind... Brilliant core course. Fun to learn with Dr Menzies. Must take for anyone aiming to be a Software Engineer. ... Good teaching skills. Enthusiastic about the course.... Brought a lot of energy to the class, and made the subject really interesting! Really liked the short language demos... Would recommend this course to anyone who asks! Wish I'd taken it last year, so I could have done some of the other course taught by Dr. Menzies..... I really loved this class because of the teaching style Dr Tim adopted, he teaches by giving these anecdotes and relating everything being taught with his real world experience. It's a joy to attend his classes and learn from him. Professor treated everyone with equality and

- took great care in making his students aware of how gender bias impact the CS world today and should be broken. He sets an example and also shows mirror to the students who are consciously/unconsciously suffering from the gender bias issues. I really want to praise and thank him for doing that. ... The instructor is enthusiastic and knowledgeable. He can explain course concepts clearly and give concrete examples in real life. After class, the instructor also happy to answer students' questions and give very useful feedback. ... After taking a really mediocre SE course in my undergrad (not at state), I had high hopes for a graduate level SE course and this definitely did not disappoint! Great instructor that made you want to come to class, with an interesting lecturing style that made me want to engage with the material.... Dr Menzies puts everyone at ease in his classes with his sense of humor. Wonderful teacher with tremendous insight and knowledge. Kept a theoretical subject like SE always interesting with good anecdotes, examples and "What did we discuss in the last class?". Liked his discussion based style of teaching.... The course is well designed, the project gives a good look-in into SE practice.
- **CSC510 (2015):**
    - Enthusiastic, Very passionate ... Dr. Menzies is a fantastic teacher. He loves research work and hard-working. ... Excellent teacher with a flair for creating interests in students .... instructors shares his own experience which is valuable for students and gives pretty good idea who software industry works. ... Assignments were good and helped in learning. ... Tim is a very good professor, whose class is full of fun. He always told something new to us. ... Awesome professor ... Amazing professor. His enthusiasm for the subject could be felt. ... Amazing professor. His enthusiasm for the subject could be felt.
  - **Csc591/791 (2014):**
    - Dr. Menzies is overall an incredibly reasonable instructor that presents material in a way that students can understand it and his expectations can be met with adequate effort. Above all his expectations are well explained at the outset of the course and if you want to be successful in the class you can be successful. ....Professor Menzies was enthusiastic, humorous, and communicated well. He used helpful visualizations, often an upside-down table or contorted elbow, as an analogue for difficult concepts.... I think the instructor is quite enthusiastic and energetic. He makes the class very much interactive.... Very enthusiastic and very intelligent. Conveys the ideas well and uses "perfect" examples .... Using the right example can save a lot of time which can be used to do more. Course was really good. Content was nice and its very relevant too. Further professor had recommended a book to follow for those who had not done much in python. He recommended it much before the classes started. I finished the book before the classes started and it helped me to become very comfortable with python as the course started. ... brilliant course ... The Instructor was well prepared and very enthusiastic.... The Instructor was well prepared and very enthusiastic.... Class is \*never\* boring; even if I had a hard time following the lectures at times, I walked away with new thoughts each class session.... Dr. Menzies is blunt with feedback but it's never personal..... Definitely the most difficult subject I've taken so far. I learned a lot and would do it again if given the choice..... This is cutting edge stuff and is worth the trouble; it's a very different course from a lot of the current offerings, however, and I would highly recommend for any prospective or current PhD students interested in research.

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

- Programming languages , (2009, 2010, 2011, 2012,2013, 2014 ), 3<sup>rd</sup> year undergraduate subject
- AI , 2011,2012,2013, 2014 4<sup>th</sup> 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), 4<sup>th</sup> year undergraduate subject
- Lightweight Software Engineering (2004), 4<sup>th</sup> year undergraduate subject
- Knowledge engineering (2002, 2003), 4<sup>th</sup> year undergraduate subject
- Software V&V (2003), Masters course year
- Modelling and analysis of software (2000), 4<sup>th</sup> year undergraduate subject
- Domain specific languages (2001), graduate class.
- OO software development (1997-98), 4<sup>th</sup> year undergraduate subject
- Visual programming (1996), 3<sup>rd</sup> year undergraduate subject
- Software engineering (1996), 3<sup>rd</sup> year undergraduate subject
- Research methods (1995,1996), 4<sup>th</sup> year undergraduate subject

## B. 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.

#### C. MENTORING ACTIVITIES

- Working with my SE faculty colleagues, developed a successful NSF Research Experience for Undergraduates grant for “Science of Software”, 2016-2018.
- Mentor for
  - Associate professor Sarah Heckman (meet, two times)
  - Assistant professor Bradley Reaves (meet twice)

#### D. MASTERS AND DOCTORAL THESES DIRECTED

I am or was chair or co-chair of the advisory committee for the following research students by thesis (students who have graduated= 7 PhD + 279MS):

Student working towards a degree (9 PhD):

1. **Amritanshu Agrawal (passed, oral prelim Jan 2019)**
2. **Jianfeng Chen (passed, oral prelim, Dec 2019)**
3. **Zhe Yu (passed, oral prelim, Dec 2018)**
4. **Rahul Krishna (passed, written prelim, Sept 2017, oral prelim scheduled Feb 2019)**
5. Tianpei (Patrick) Xia
6. Huy (Ken) Tu
7. Suvodeep Majumder
8. Joymallya Chakraborty
9. Rui Shu
10. Fahmid Fahid
11. Shrikanth Chandrasekaran
12. Xueqi (Sherry) Yang

Completed Ph.D.:

1. **Vivek Nair, (2019, NCSU) Frugal Ways to Find Good Configurations.**
2. **Wei Fu, (2018, NCSU) Simpler Software Analytics: When? When Not?**
3. Abdel Sayyad Ph.D. (2014, WVU) *Evolutionary Search Techniques with Strong Heuristics for Multi-Objective Feature Selection in Software Product Lines*
4. Joe Krall Ph.D. (2014, WVU) *Active Learning for Search-Based Software Engineering*
5. Fayola Peters Ph.D. (2014, WVU) *Privacy and Data Sharing*
6. Ekrem Ph.D. (2012, WVU) *A Principled Methodology: A Dozen Principles of Software Effort Estimation*
7. Nandeshwar, Ashutosh Ph.D. (2011, WVU) *Longitudinal study of first-time freshmen using data mining*

8. David Owen Ph.D. (2010, WVU) *Combining complementary formal verification strategies to improve performance and accuracy*
9. Scott Chen Ph.D. (2004, U.Sc.) *Data Mining for Effort Estimation*

Completed Masters:

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

### III. SCHOLARSHIP IN THE REALMS OF FACULTY RESPONSIBILITY

#### A. SCHOLARLY ACCOMPLISHMENTS – PUBLICATIONS

*Submitted (under review)*

---

**3\* EMSE journal**  
**1\* ICPC conference**  
**3\* TSE journal**  
**1\*IST Journal**

#### *Invited and Contributed Research Presentations*

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

#### *Refereed Journal and Top Magazine Publications*

- 1. Vivek Nair, Zhe Yu, Tim Menzies, Norbert Siegmund, Sven Apel, Finding Faster Configurations using FLASH, accepted IEEE Transactions on SE, 2018**
- 2. G Mathew, A Agrawal, T Menzies. Finding Trends in Software Research, IEEE Transactions on Software Engineering, 2018 accepted for publication.**
- 3. 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**
- 4. Zhe Yu, Nicholas A. Kraft, Tim Menzies, Finding better active learners for faster literature reviews. Empirical Software Engineering 23(6): 3161-3186 (2018)**
- 5. 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)**
- 6. Z Yu, T Menzies. , FAST2: An intelligent assistant for finding relevant papers Expert Systems with Applications 120, 57-71, 2019**
7. J Nam, W Fu, S Kim, T Menzies, L Tan, Heterogeneous defect prediction, IEEE Transactions on Software Engineering, June 2017
8. J Chen, V Nair, R Krishna, T Menzies , "Sampling" as a Baseline Optimizer for Search-based Software Engineering , IEEE Transactions Software Engineering, accepted 2017, to appear.
9. V Nair, T Menzies, N Siegmund, S Apel, Faster discovery of faster system configurations with spectral learning, Automated Software Engineering, 30 August 2017.

10. J Chen, V Nair, T Menzies, Beyond Evolutionary Algorithms for Search-based Software Engineering, Information and Software Technology Accepted 2017, to appear.
11. Rahul Krishna, Tim Menzies, Lucas Layman: Less is more: Minimizing code reorganization using XTREE. Information & Software Technology 88: 53-66 (2017)
12. 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. To appear.
13. T Menzies, Y Yang, G Mathew, B Boehm, J Hihn, , Negative results for software effort estimation Empirical Software Engineering, 1-26, 3, 2016
14. 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.
15. 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
16. 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.
17. 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
18. Transfer learning in effort estimation, E Kocaguneli, T Menzies, E Mendes Empirical Software Engineering, 1-31, 2014
19. SN Partington, V Papakroni, T Menzies , Optimizing data collection for public health decisions: a data mining approach, BMC Public Health 14 (1), 593, 2014
20. 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
21. 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
22. 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
23. 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)
24. 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)
25. 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
26. 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)
27. "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
28. 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 ,
29. 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,
30. 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)
31. 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)
32. "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)
33. 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
34. 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
35. 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.
36. "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 .



37. "What is Enough Quality for Data Repositories?" by Tim Menzies and Adam Brady and Ekrem Kocaguneli. Software Quality Professional, volume 13, number 3, 2011 .
38. 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> .
39. 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> .
40. 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> .
41. 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> .
42. 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> .
43. Oussama El-Rawas and Tim Menzies, "A Second Look at Faster, Better, Cheaper" in. Innovations Systems and Software Engineering pages 319-335 2010 .
44. 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> .
45. 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> .
46. 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> .
47. 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>
48. 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.
49. 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>.
50. 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>.
51. 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>.
52. 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>.
53. 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>.
54. 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>.
55. T. Menzies, D. Owen, and J. Richardson. The strangest thing about software. IEEE Computer, 2007. <http://menzies.us/pdf/07strange.pdf>.
56. 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>.
57. T. Menzies and J. Richardson. Making sense of requirements, sooner. IEEE Computer, October 2006. Available from <http://menzies.us/pdf/06qrre.pdf>.
58. 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>.
59. 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>.
60. 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>.
61. Zhihao Chen, Tim Menzies, Dan Port, and Barry Boehm. Finding the right data for software cost modeling. IEEE Software, Nov 2005.
62. 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>.
63. 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>.
  64. T. Menzies and Y. Hu. Data mining for very busy people. In IEEE Computer, November 2003. Available from <http://menzies.us/pdf/03tar2.pdf>.
  65. 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>.
  66. T. Menzies and B. Cukic. When to test less. IEEE Software, 17(5):107–112, 2000. Available from <http://menzies.us/pdf/00iesoft.pdf>.
  67. 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>.
  68. 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>.
  69. 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>.
  70. 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>.
  71. 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>.
  72. T. Menzies. Cost benefits of ontologies. ACM SIGART Intelligence magazine, Fall 1999. Available from <http://menzies.us/pdf/99sigart.pdf>.
  73. 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>.
  74. 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>.
  75. 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>.
  76. 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>.
  77. T.J. Menzies. An investigation of the ai and expert systems literature 1980-1984. AI Magazine, Summer 1989.
  78. 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/ooovis95.pdf> .

#### Refereed Conference Publications

1. 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
2. Di Chen, Wei Fu, Rahul Krishna, Tim Menzies: Applications of psychological science for actionable analytics. ESEC/SIGSOFT FSE 2018: 456-467
3. 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
4. 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
5. 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
6. Vivek Nair, Amritanshu Agrawal, Jianfeng Chen, Wei Fu, George Mathew, Tim Menzies, Leandro L. Minku, Markus Wagner, Zhe Yu:
7. Data-driven search-based software engineering. MSR 2018: 341-352
8. Jianfeng Chen, Tim Menzies: RIOT: A Stochastic-Based Method for Workflow Scheduling in the Cloud. IEEE CLOUD 2018: 318-325
9. Chin-Jung Hsu, Vivek Nair, Tim Menzies, Vincent W. Freeh: Micky: A Cheaper Alternative for Selecting Cloud Instances. IEEE CLOUD 2018: 409-416
10. 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
- Wei Fu, Tim Menzies: Easy over hard: a case study on deep learning. ESEC/SIGSOFT FSE 2017: 49-60
11. Wei Fu, Tim Menzies: Revisiting unsupervised learning for defect prediction. ESEC/SIGSOFT FSE 2017: 72-83
12. Vivek Nair, Tim Menzies, Norbert Siegmund, Sven Apel: Using bad learners to find good configurations. ESEC/SIGSOFT FSE 2017: 257-267
13. George Mathew, Tim Menzies, Neil A. Ernst, John Klein: "SHORT"er Reasoning About Larger Requirements Models. RE'17: 154-163
14. 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
15. 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.
16. R Krishna, T Menzies, W Fu, Too much automation? the bellwether effect and its implications for transfer learning, ASE'16.
17. J Hihn, L Juster, J Johnson, T Menzies, G Michael, Improving and expanding NASA software cost estimation methods Aerospace Conference, 2016 IEEE, 1-12
18. 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)
19. Jairus Hihn, Tim Menzies, Improving and Expanding NASA Software Cost Estimation Methods 2016 IEEE Aerospace Conference.
20. Scalable product line configuration: A straw to break the camel's back, ASE , 2013 , AS Sayyad, J Ingram, T Menzies, H Ammar
21. 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
22. Class level fault prediction using software clustering, G Scanniello, C Gravino, A Marcus, T Menzies, ASE 2013
23. 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

24. 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
25. Ekrem Kocaguneli, Bojan Cukic, Tim Menzies, Huihua Lu: Building a second opinion: learning cross-company data. PROMISE 2013: 12
26. Beyond data mining; towards idea engineering T Menzies, PROMISE 2013
27. Ekrem Kocaguneli, Thomas Zimmermann, Christian Bird, Nachiappan Nagappan, Tim Menzies: Distributed development considered harmful? ICSE 2013: 882-890
28. Fayola Peters, Tim Menzies: Privacy and utility for defect prediction: Experiments with MORPH. ICSE 2012: 189-199
29. Yang Sok Kim, Byeong Ho Kang, Seung Hwan Ryu, Paul Compton, Soyeon Caren Han, Tim Menzies: Crowd-Sourced Knowledge Bases. PKAW 2012: 258-271
30. Raymond Borges, Tim Menzies: Learning to change projects. PROMISE 2012: 11-18
31. Ekrem Kocaguneli, Tim Menzies, Jaius Hihn, Byeong Ho Kang: Size doesn't matter?: on the value of software size features for effort estimation. PROMISE 2012: 89-98
32. "How to Find Relevant Data for Effort Estimation?" by Kocaguneli, E. and Menzies, T.. Proceedings ESEM11, 2011
33. "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> .
34. "Text mining in supporting software systems risk assurance" by Huang, LiGuo and Port, Daniel and Wang, Liang and Xie, Tao and Menzies, Tim. IEEE ASE10 pages 163--166 2010. Available from <http://menzies.us/pdf/10textrisk.pdf> .
35. "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> .
36. "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> .
37. "Software is Data Too" by A. Marcus and T. Menzies. FoSER 2010 November 2010 . Available from <http://menzies.us/pdf/10softwareisdata.pdf> .
38. "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> .
39. "Regularities in Learning Defect Predictors" by Burak Turhan, Ayse Bener and Tim Menzies. Profes 2010 2010 . .
40. "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> .
41. 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> .
42. 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> .
43. 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,
44. 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> .
45. 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> .
46. 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> .
47. 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> .
48. 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> .
49. 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> .
50. 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> .
51. 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> .
52. T. Menzies and A. Marcus. Automated severity assessment of software defect reports. In ICSM'08, 2008. Available from <http://menzies.us/pdf/08severis.pdf> .
53. 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> .

54. 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>.
55. 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>.
56. T. Menzies, O. Elrawas, J. Hihn, M. Feather, 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.
57. 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>.
58. 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>.
59. 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>.
60. 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>.
61. 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>.
62. M.S. Fisher and T. Menzies. Learning ivv strategies. In HICSS'06, 2006. Available from <http://menzies.us/pdf/06hicss.pdf>.
63. 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](http://menzies.us/pdf/05xomo_cocomo_forum.pdf). For more details, see also the longer technical report <http://menzies.us/pdf/05xomo101.pdf>.
64. 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>.
65. 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>.
66. 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>.
67. D. Owen and T. Menzies. Lurch: a lightweight alternative to model checking. In SEKE '03, 2003. Available from <http://menzies.us/pdf/03lurch.pdf>.
68. 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>.
69. 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>.
70. 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>.
71. 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>.
72. 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>.
73. 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>.
74. 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>.
75. 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>.
76. 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>.
77. 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>.

78. 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>.
79. 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>.
80. 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>.
81. 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>.
82. 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>.
83. 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>.
84. 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>.
85. 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>.
86. T. Menzies and J.D. Kiper. Better reasoning about software engineering activities. In ASE-2001, 2001. Available from <http://menzies.us/pdf/01ase.pdf>.
87. Tim Menzies, Bojan Cukic, Harhsinder Singh, and John Powell. Testing nondeterminate systems. In ISSRE 2000, 2000. Available from <http://menzies.us/pdf/00issre.pdf>.
88. 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>.
89. 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>.
90. 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>.
91. 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.
92. 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>.
93. 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>.
94. 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>.
95. 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.
96. 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>.
97. S Ramakrishnan and T. Menzies. An ongoing experiment in o-o software process and product measurements. In Proceedings SEEP'96, New Zealand, 1996.
98. 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>.
99. T.J. Menzies. On the practicality of abductive validation. In ECAI '96, 1996. Available from <http://menzies.us/pdf/96ok.pdf>.
100. 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>.
101. 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>.
102. 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.

103. 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>.
104. 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.
105. T.J. Menzies and R. Spurrett. How to Edit “t” 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>.
106. P. Haynes and T.J. Menzies. C++ is Better than Smalltalk? In Tools Pacific 1993, pages 75–82, 1993.
107. 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>.
108. 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>.
109. 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>.
110. 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.
111. Parametric analysis of a hover test vehicle using advanced test generation and data analysis.
112. 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>.
113. S. Waugh, J. Blogs, and T. Menzies. The temporal qualitative compartmental modeling language. In Proceedings of the Australian AI '98 conference, 1998. Available from <http://menzies.us/pdf/97links.pdf>.
114. 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>.
115. 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>.
116. 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>.
117. 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>.
118. 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>.
119. T.J. Menzies. Maintaining procedural knowledge: Ripple-down-functions. In Proceedings of AI '92, Australia, 1992. Available from <http://menzies.us/pdf/ai92.pdf>.
120. 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.
121. T.J. Menzies. Isa object part-of knowledge representation? In Proceedings AI '90, 1990.
122. 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.
123. T.J. Menzies and C. Worral. Worlds in prolog. In Proceedings of AI '87, 1987.
124. 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/99ekeepdf.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/96ecaimc.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>.

## B. RESEARCH FUNDING

New for this year: \$308,628

Accepted:

- LAS: How Safe is this conclusion?: \$35,000
- Lexis Nexis: Entity Recognition: \$40,000

- Lexis Nexis: configure + cloud tests \$110,000
- SHF: EAGER: Empirical SE for Computational Science: \$124,628

Under review:

Declined:

- SHF: Small: Jumpstarting Next Generation Vulnerability Prediction Models, \$499,857
- SHF: Medium: Better Software Analytics by Combining Human and Artificial Intelligence \$905037

						grant: sole	co-PI	Research	Grants-	Research		
						PI	(expended by	expenditure	Co-PI	expenditure		
							TM: )		(total)			
start	finish	Funding body	Name	gift?	A	B	C=A+B	D	E=C+D	current?	new for	sum for
											may2017-	may2017-
											april2018	april2018
2018	2018	LAS (NSA)	How to make a magican		\$35,000		35,000		\$35,000	yes	yes	
2017	2021	NSF	Autotuning		\$0	\$450,000	450,000	\$450,000	\$900,000	yes	yes	
2017	2017	IBM	Automated SE: Faculty grant	y	\$40,000		40,000		\$40,000	yes	yes	\$975,000
2015	2018	Lexis Nexis (Raleigh)	Validation lab	y	\$120,000		120,000		\$120,000	yes		
2017	2017	Lexis Nexis (Atlanta)	Understanding policies	y	\$35,000		35,000		\$35,000	yes		
2017	2017	Lexis Nexis (Atlanta)	The agreement machine	y	\$35,000		35,000		\$35,000	yes		
2017	2017	LAS (NSA)	Privitized data sharing		\$35,000		35,000		\$35,000	yes		
2016	2016	IBM	Automated SE: Faculty grant	y	\$40,000	\$0	40,000	\$0	\$40,000	yes		
2016	2018	NSF	Reu: Science of Software			\$10,000	10,000	\$345,365	\$355,365	yes		
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	NCDISA	Share Care Beware		\$60,000	\$0	60,000	\$0	\$60,000	yes		
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	yes		
2013	2014	NASA (JPL)	Effort estimation		47,000		47,000		\$47,000			
2012	2016	USDA	Early Childhood Obesity Program		\$133,526		\$133,526		\$133,526			
2012	2013	NSF	New directions in AI and SE		\$14,700		\$14,700		\$14,700			
2010	2012	Dod STTRv	Active Learning		\$230,514		\$230,514		\$230,514			
2010	2014	NSF (CISE)	Better Understanding of SE data			\$249,500	\$249,500	\$499,000	\$748,500			
2010	2012	Qatar Resarch	Int Center of Excellence in SE			\$98,125	\$98,125	\$196,250	\$294,375			
2010	2011	CITRE	Border Crossing		\$70,000		\$70,000		\$70,000			
2010	2011	National Forensics	Overcoming Brittleness		\$35,721		\$35,721		\$35,721			
2009	2010	National Archives	STEP Research			\$209,000	\$209,000	\$418,000	\$627,000			
2008	2009	National archives	STEP research			\$143,500	\$143,500	\$574,000	\$717,500			
2008	2011	NSF (CISE)	Automatic Quality Assessment			\$180,000	\$180,000	\$360,000	\$540,000			
2008	2009	National Forensics	Conclusion stability		\$80,000		\$80,000		\$80,000			
2008	2008	NASA	Understanding Anomalies.		\$58,000		\$58,000		\$58,000			
2008	2008	NASA	Crystal Ball.		\$55,000		\$55,000		\$55,000			
2008	2008	NASA	Advanced UML modeling.		\$50,000		\$50,000		\$50,000			
2007	2008	NASA	Applied Technology Lab		\$95,551		\$95,551		\$95,551			
2007	2008	Dod STTRv	Next generation metrics: phase 1		\$40,715		\$40,715		\$40,715			
2007	2007	NASA	WVU Liaison		\$39,707		\$39,707		\$39,707			
2007	2008	Industrial	Analysis metrics (Galaxy Global)		\$25,000		\$25,000		\$25,000			
2007	2008	National archives	STEP research		\$15,482		\$15,482		\$15,482			
2006	2007	NASA	Learning software process model		\$113,255		\$113,255		\$113,255			
2006	2007	NASA	Improving IV&V Techniques		\$107,990		\$107,990		\$107,990			
2006	2006	NASA	co-op agreement supplemental funds			\$14,916	\$14,916	\$59,665	\$74,581			
2006	2006	NASA	co-op funds for Eislund Hall Lab		\$30,000		\$30,000		\$30,000			
sum since tenure (2006+)					\$1,932,161	\$1,977,071	\$3,909,232	\$3,432,051	\$7,376,285			
2005	2005	NASA	How to Argue Less:		\$260,000		\$260,000		\$260,000			
2005	2005	NASA	Spectrum of Model Checking Methods		\$160,000		\$160,000		\$160,000			
2005	2005	NASA	Risk/Cost models for Autonomy		\$160,000		\$160,000		\$160,000			
2005	2005	NASA	How much will it cost?		\$122,161		\$122,161		\$122,161			
2005	2005	NASA SBIT	Intelligent Vehicle Health Management:		\$65,000		\$65,000		\$65,000			
2004	2004	NASA	Spectrum of Model Checking Methods		\$160,000		\$160,000		\$160,000			
2004	2005	NASA	A next-generation testable language		\$70,000		\$70,000		\$70,000			
2004	2004	NASA	The research rover		\$48,000		\$48,000		\$48,000			
2003	2005	NASA	Understanding models better		\$107,000		\$107,000		\$107,000			
2003	2003	NASA	Model checking & procedural languages		\$50,000		\$50,000		\$50,000			
2003	2003	NASA	See more! Learn more! Tell more!		\$47,000		\$47,000		\$47,000			
2002	2003	NASA	A spectrum of IV&V techniques		\$200,000		\$200,000		\$200,000			
2002	2002	NASA	Better risk modelling		\$27,000		\$27,000		\$27,000			
2001	2001	NASA	Tree query languages		\$27,000		\$27,000		\$27,000			
2000	2000	Canada Res. Coun.	NSERC grant		\$81,000		\$81,000		\$81,000			
1998	1999	NASA	High Quality Knowledge Initiative		\$110,000		\$110,000		\$110,000			
1997	1998	Aust. Res. Coun	Abduction for software engineering		\$10,000		\$10,000		\$10,000			
1996	1998	UNSW	Vice-Chancellor's Research Fellowship		\$135,000		\$135,000		\$135,000			
Total (1996 to 2005)					\$1,839,161	\$0	\$1,839,161	\$0	\$1,839,161			
Total (ALL)					\$3,771,322	\$1,977,071	\$5,748,393	\$3,432,051	\$9,215,446			

- HERE IS THE NSCU RADAR REPORT. NOTE THAT IT DOES NOT INCLUDE
  - MY GIFT AMOUNTS
  - OR THE \$624K IN NSF PROPOSALS THAT ARE CURRENTLY UNDER REVIEW

## NORTH CAROLINA STATE UNIVERSITY

**Proposals Submitted for the Period January 01, 2017 to February 02, 2018 by College and Department**

**investigator is Menzies, Timothy; college is COLLEGE OF ENGINEERING (PI college only); department is Computer Science (PI department only); revised proposals suppressed; pre-proposals suppressed; internal projects suppressed**

Project ID	Investigator(s)	SP	PI Department	Direct Sponsor	Amount	Type	Category	Title
COLLEGE OF ENGINEERING								
<a href="#">2018-0729</a>	Menzies, Timothy		Computer Science	National Science Foundation (NSF)	\$905,037	New	Research	SHF: Medium: Better Software Analytics by Combining Human and Artificial Intelligence
<a href="#">2018-1293</a>	Menzies, Timothy Williams, Laurie		Computer Science	National Science Foundation (NSF)	\$499,857	New	Research	SHF: Small: Jumpstarting Next Generation Vulnerability Prediction Models
Computer Science (141901):					\$1,404,894	2		
COLLEGE OF ENGINEERING Total:					\$1,404,894	2		
Grand Total:					\$1,404,894	2	2 projects	

[Return to Report Menu](#)

#### C. CENTERS AND MULTIDISCIPLINARY COLLABORATION

N/A

#### IV. EXTENSION AND ENGAGEMENT WITH CONSTITUENCIES OUTSIDE THE UNIVERSITY

- **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

#### V. 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

For my full career impact, see section B2

B1. IMPACT, last 12 months

1. *What are the highlights of your achievements in the past year?*

2.

- One of my 2007 papers have now past 1000 citations
- At the time of this writing, at IEEE transactions on SE, three of the top twenty most cited papers are by me and my Students (Google Scholar, Software Systems).
- The “Bellwether” research is not getting cited in multiple papers from other researchers
- Elevated to IEEE Fellow, which is personally gratifying but also has some useful benefits to the university for reputational purposes
- As part of the IEEE fellow evaluation process, got to revisit some old work. I found that much of old NASA research continued even after I left:
  - In 2005, my 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 my optimizers to guide discussions about costly updates to Department of Defense software.
  - NASA used my 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;
    - 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.

2. *In the last year, how have you had impact? Who have you impacted?*

- Working with Lexis Nexis,
  - Reorganized the data collection for their cloud testing environments such that they were better suited to assessing test effectiveness.
- 

3. *In the last year, what have you done to help the department:*

- Recruit and retain faculty of the highest caliber.
  - Engage with other faculty outside of normal work activities.
    - Served as a committee member for a English student's Ph.d. (Peter Kudenov).
  - Search committees
    - Served on hiring committees

- Meet with numerous candidates as part of their on-site interviews
- *I serve as mentor for*
  - Associate professor Bradley Reeves
  - Associate professor Sarah
- *Working with other senior colleagues on grants*
  - I co-wrote an NSF small grant with full professor Laurie Williams
- 
- *What have you done to promote your press coverage?*
  - Very active on social media
  - **Researchgate.com reports that I was the most cited author in the CSC department for much of 2018.**
- *Recruit and retain high-quality PhD students.*
  - **Worked with the SE faculty to select and target and contact exceptional Ph.D. candidates**
  - Worked with the CSC grad office for the recruiting weekend. I devoted 2 days to that project- organized presentations for the students interested in SE; spent some one-on-one time with various candidates, attended the evening banquet, advised the CSC grad office about which candidate might be worthy of additional financial support.
  - As to retaining grad students, I've created and maintained extensive relationships with Lexis Nexis and ABB in order to offer students interesting big data projects.
  - I also supervise a large number of Ph.D. students (nine)
- *Increase multi-disciplinary research collaborations, including sustainable centers of excellence.*
  - I am now on the committee for an English digital Humanities Ph.D. candidate (Peter Kudenov)
- *Encourage active collaboration with business, government, and other universities*
  - I am a consultant to IBM research triangle regarding their analytics program.
- *Expand collaboration and outreach to K-12 students and teachers*
  - I worked last summer on the SE NSF REU project. Supervised 3 students. Gave some guest lectures.
- *Offer outstanding and current curricula at the undergraduate and graduate levels*
  - **I developed a new subject "Foundations of Software Science" <http://git.io/fss18>**
- *Creation or used a repository of course material available to other faculty*
  - All my course material is on-line, available for download via Github
  - E.g. Teaching materials for Model-based Automated Software Engineering <http://github.com/txt/ase16>
  - E.g. Teaching materials for Foundations of Software Science: <http://git.io/fss18>
  - e.g. Complete notes/ project definitions for Graduate SE: <https://github.com/txt/se16>
- *Incorporated into classes of team projects, written reports, in-class presentations*
  - All my CSC 510 students report back to the class in marathon 3 week-long presentation at end of semester.

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

- Cover my cloud compute costs (not HPC, but ARC)
- Better ways to select better Ph.D. students
- More support go Grad student travel

5. *What are your goals for the 2017-2019 year?*

- More publications for my graduate students
- More long term funding for my graduate students

## B2. IMPACT, full career

**Optimization of Software-Intensive Systems :** *I have developed (and demonstrated the value of) very fast non-numeric optimizers for software-intensive systems.* For the purposes of controlling spacecraft re-entry guidance software, my learners generated better controllers and ran 40 times faster than the state- of-the-art numeric optimizers<sup>4</sup>.

- For the purposes of extracting products from product lines, for 2013,2014, my algorithms represented the state of the art on that field (most number of goals, largest models<sup>5</sup>).
- **In 2017, the US Software Engineering Institute used my optimizers to guide discussions about costly updates to Department of Defense software.**
- **NASA has used my 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;**
  - **in 2017 to find better monitoring strategies for pilots flying planes in safety-critical situations.**

**Software Defect Prediction:** *I have been instrumental in the development of experimental methods that allow for the discovery and precise evaluation of software defect predictors generated by data miners from static code attributes.*

According to Martin Shepperd my IEEE TSE 2007<sup>6</sup> paper on learning defect predictors “represents the state of the art for this kind of research” and “is widely cited”.

- This work is the most cited paper 2005 to 2014 in IEEE Transactions in Software Engineering (source: <http://academic.research.microsoft.com>).
- **This work is now listed as being amongst of the 16<sup>th</sup> most most-cited SE papers of all time<sup>7</sup>.**
- **In 2005, my 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 2009, researchers in Turkey demonstrated the effectiveness of my methods in a commercial setting. This is one of the first and the most prominent demonstrations of the value of these methods.**

**Cross-company Learning :** *I was one of the first to demonstrate that effective local effort/defect models can be learned from data imported from other companies. I have shown that useful models can be built if relevancy filtering selects the subset of the data from other companies that is most relevant to the local company<sup>8</sup>.*

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

**Software Effort Estimation :** *I remain one of the most prominent publishing researches in this field.* Elsewhere, I have addressed, and reduced, one of the major outstanding problems in the field of effort estimation; *i.e conclusion instability*. Using Ensemble learning, I have found that if we study enough data sets and enough learners then the magnitude of that instability is less than the total sample<sup>9</sup>.

- Another 2012 paper of mine on analogy and effort estimation as the most-cited estimation paper in the last five years<sup>10</sup> (source: Google Scholar metrics)

<sup>3</sup> Tosun, Ayşe, et al. "Practical considerations in deploying statistical methods for defect prediction: A case study within the Turkish telecommunications industry." *Information and Software Technology* 52.11 (2010): 1242-1257.

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

<sup>5</sup> Scalable product line configuration: A straw to break the camel's back, ASE , 2013 , AS Sayyad, J Ingram, T Menzies, H Ammar

<sup>6</sup> 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>.

<sup>7</sup> <http://menzies.us/pdf/mostCitedSEpapers.pdf>

<sup>8</sup> 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>.

<sup>16</sup> 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)

<sup>10</sup> Kocaguneli, E.; Menzies, T.; Keung, J.; Cok, D.; Madachy, R.; , "Active Learning and Effort Estimation: Finding the Essential Content of Software



- In 2016, based on my research, NASA's Jet Propulsion Laboratory created the NASA Analogy Software Cost Model as its official tool for predicting software development costs.

**Software Requirements Engineering:** *I was one of the earliest pioneers in the field of search-based software engineering for requirements engineering.* According to Mark Harman, in 2002 I was one of the earliest to apply Pareto optimality in search-based software engineering (SBSE) for requirements engineering<sup>11</sup>

**Measurement Errors :** *I have discovered a previously undocumented subtle, and dangerous, aspect of a widely-used performance measure.* Precision is a commonly-used assessment measure used in data mining. In 2007, I showed that this seemingly simple measure had significant problems when the target class is relatively rare (specifically, for such data sets, seemingly minor changes in the learning process can lead to massive changes in the precision values)<sup>12</sup>.

**The PROMISE Project:** *I have been very active in solving an outstanding problem in software engineering: accessing the data required for repeatable experiments.* In order to support the above research, I had to create a source of SE data. Accordingly, in 2005, I founded the PROMISE conference on repeatable experiments in software engineering. The goal of PROMISE is to give the research community free access to the data sets that we can use for analysis of software engineering data. That work recently won me the inaugural Mining Software Repositories Foundational Contribution Award. See <http://2017.msrf.org/#/award>.

The following notes come from that award. In this era of Github, GHTorrent, et al. it is hard to recall that only a decade ago, it was difficult to access project data. Nevertheless, that was the case. Back in 2005 many people in the MSR field were analyzing large amount of (public) open source data but kept the tools and processed datasets to themselves as it was often considered a competitive advantage. In fact, within the MSR community, it was not until 2013 that they started their Data Showcase track to encourage sharing of data. Meanwhile, back in 2005, I started the PROMISE workshop to encourage data sharing.

The following notes come from reviewers who endorsed by nomination for this award:

- "I don't think any other contribution that is even remotely comparable to (PROMISE)."
- "There are many factors that lead to the state-of-the-art in mining software repositories, ... sharing and replicability in the science of software engineering, but in my opinion, none played a greater role than the PROMISE repository, created by Tim Menzies."

---

Effort

Estimation Data," Software Engineering, IEEE Transactions on ,

<sup>9</sup> 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>.

<sup>12</sup> 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>.

**NC STATE UNIVERSITY**

**Computer Science**

RESOURCES search csc.ncsu.edu

About Academics Research People News Contact

**CSC News**

April 24, 2017

**Menzies' PROMISE Earns Him International Accolades**

Dr. Timothy Menzies, Professor of Computer Science at N.C. State, has been recognized for his major contributions in the field of mining software repositories. Menzies is the curator of the **PROMISE** repository, storage for Software Engineering project data.

Menzies' "PROMISE project" first began in 2005, when Menzies and his students couldn't get the data they needed for their work. Seeing this issue, the team went out to search for the data they needed. This project Menzies and his team first began in their spare time, quickly developed into an internationally recognized project, challenging the status-quo of data collection. Menzies explains just how much data collection has changed in this recent past, and holds onto the idea that sharing information leads to future successes in any field.

"In this age of connected information it may be a surprise to learn that there was a time where people just didn't share data. Our thought process was: If you have anything, give it away."

The project is composed of two parts: a repository to store data from papers from the conference, and an annual conference called "PROMISE" that's currently co-located with ESEM (Empirical Software Engineering and Measurement). Both parts have created hundreds of papers, using dozens of students, resulting in some of the most cited papers within the field of Software Engineering.

Not only has the project made innumerable strides in the field of data collection, but thousands of connections have been made with students and the working world within their individual fields of study. Menzies remembers students studying abroad in Beijing, connecting with leaders in the industry on a Global level, and receiving valuable experience in their field of study.

Menzies will be receiving his award as part of the May 2017 International Conference on Software Engineering. For more details, see <http://menzies.us/msraward17.html>.

- "I can't think of a stronger contribution right now (to MSR), except maybe the creation of the MSR community by Dr. Hassan."
- "PROMISE makes research results reproducible. For example, according to Robles et al. at MSR 2010, they found that over 95% of 171 papers published at MSR were unreproducible, since their associated data was no longer online. However, nearly all of the papers at the PROMISE conference was reproducible since they were based on PROMISE data. I believe this is a really significant contribution."
- "Prof. Menzies has enabled and inspired a generation of SE data scientists to use data to make refutable results. In his own publications, he makes extensive use of PROMISE data with the express aim of challenging others to share their data and/or repeat/improve/ or refute his own results. For example, his 2007 TSE article "Data Mining Static Code Attributes to Learn Defect Predictors" contains numerous interesting research contributions. But much of the paper reads like a "how to" guide on software analytics. All its data was placed online at PROMISE. This paper is widely cited, to say the least: 770 citations since 2007; most cites per year for any TSE paper since its publication; one the top100 most cited papers anywhere in the history of SE (see <http://dl.acm.org/citation.cfm?id=2874005>). That paper ends with following paragraph:

*Our hope is that numerous researchers repeat our experiments and discover learning methods that are superior to the one proposed here. Paradoxically, this paper will be a success if it is quickly superseded.*

I find this paragraph inspirational, a shining example of how to do open science in the Internet age. Here Dr. Menzies is pleading with the international research community to outperform his own publication. Further, by placing all the materials for this paper online at PROMISE, he is enabling other researchers to refute his work, if they can."

## VI. SERVICE TO THE UNIVERSITY AND PROFESSIONAL SOCIETIES

### A. UNIVERSITY SERVICE

- **Cahir, Search committee, SE faculty, 2019**
- **Member, CSC Graduate Program Oversight Committee. 2016-**
- **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

- 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:
  - **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:
  - **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, Icpc'15, Issre'15, Msr'15, NasBase'15, Promise'15, Raise'15, Ssbse'15
  - 2014:
    - MSR'14, ICSE14-demos, ICSE14-mainConference, DAPSE'14, EASE'14, GTSE'14, SAM 2014, SEAA 2014,
  - Before 2014:
    - Mining Software Engineering 2013, 2012, '2011

- IEEE Automated Software Engineering (2013,2012,2011,2010,2009, 2008,2007,2005, 2004, 2003, 2002)
  - Empirical Software Engineering and Measurement '2012 '2011, 2013
  - SAM2103,
  - DAPSE'13
  - ICSE'13: demos
  - ASE-Tools'13
  - ISSRE'13
  - GTSE'13
  - MALIR'13
  - Software Mining -2012, 2013
  - RAISE'12, RAISE'13
  - FSE New ideas'11,
  - Software engineering week, 2011,
  - Spark'11
  - IEEE International Symposium on Software Reliability Engineering (2010,2009);
  - Pacific Knowledge Acquisition Workshop, 2009,2008
  - LSO (learning software organizations), 2008
  - Traceability in Emerging forms of SE , 2007
  - International Workshop on Living with Uncertainty (2007)
  - IEEE conference on high assurance software engineering (2007, 2004);
  - 17th International Conference on Automated Planning & Scheduling (2007)
  - MoChArt '05 (model checking and AI)
  - Tim Menzies, vita page 7 of 23
  - IEEE NASA Software Engineering Workshop (2003)
  - IEEE Metrics 2003;
  - Numerous other PCs since 1991 including
    - 8 international conferences
    - 16 international workshops,
    - 5 Australian national workshops.
    - Organizing committee member for 2 international workshops, 4 national conferences and workshops.
- Reviewer for:
    - ACM Transactions on Software Engineering and Methodology, IEEE Transactions on Software Engineering, Empirical Software Engineering, Automate Software Engineering, Information Systems and Technology, Applied Soft Computing, IEEE Software, International Journal of Human Computer Studies. Software Quality Journal, Software Process: Improvement and Practice Journal, Software Testing, Verification, and Reliability , IEEE Transactions on Evolutionary Computation