NOMINATION PACKET

State Council of Higher Education for Virginia

2019 Outstanding Faculty Award

Jeff Offutt, PhD

Professor at George Mason University

Table of Contents

Section B. George Mason University's Mission Statement

George Mason University – A university for the world

Mission – Who we are and why we do what we do

A public, comprehensive research university established by the Commonwealth of Virginia in the National Capital Region, we are an innovative and inclusive academic community committed to creating a more just, free, and prosperous world.

The Mason *idea*—Our core institutional characteristics

- **Innovative** = We do not cling to old ways just because they have worked in the past. We honor time-tested academic principles, while we strive to create new forms of education that serve our students better and new paths of research that can help us discover solutions to the world's greatest challenges.
- **Diverse** = We bring together a multitude of people and ideas in everything that we do. Our culture of inclusion, our multidisciplinary approach, and our global perspective make us more effective educators and scholars.
- **Entrepreneurial** = We take ideas into action. We educate students to become agents of positive change; to do or create jobs; to create value through government or business, public or private organizations, academia, or the arts. We pursue discoveries that can make a difference in the world. We help our community thrive culturally, socially, and economically.
- **Accessible** = We are an open and welcoming community. We partner with public and private organizations in our region and around the world. We proactively engage with our community. We define our success by how many talented students with potential we serve, not how many we leave out.

Section C. Summary of Accomplishments of Professor Jeff Offutt

As a child, Offutt did not seem destined for academic success. He was a desultory and disrespectful student, unhappy with the slow pace and lack of challenge. The dearth of educational opportunities in Appalachian Kentucky left him hungry for knowledge. He got two breaks late in college that changed his life. First, he landed a summer internship at the prestigious Exxon Research Labs. Second, a great graduate school focused on GRE scores, not grades. At Georgia Tech, he absorbed computer science like a sponge. This path left him with a unique attitude towards his profession—instead of separating knowledge discovery, teaching, and service, Offutt sees them as one thing. As knowledge. Finding new knowledge, passing it on through teaching, and ensuring that others have opportunities through his service.

Offutt is a world-renowned software engineering educator and researcher. He has published over 180 fully refereed research papers, has an h-index[1] of 61 (Google Scholar), and has received funding from many government agencies and companies. These include the NSF, Navy, NASA, DARPA, DHS, Google, Freddie Mac, Saab Aerospace, Avaya Labs, NIST, and Rockwell-Collins. With frequent collaborator Dr. Paul Ammann, he has also written the most widely used textbook in software testing, with over 75% of the worldwide market. He has created and taught many courses at all levels of collegiate education, from freshman to PhD students. He has published more than a dozen education papers and his course materials are given freely and used around the world. He learns through his research, he researches better teaching methodologies and teaches research. He uses service to improve teaching and research.

DISCOVERY AND RESEARCH. Professor Offutt's research has influenced the fields of software testing and software engineering in major ways.

- His inventions are relevant, and many are in practical use at software companies, including mutation testing, bypass testing, base choice testing, data perturbation, prime path testing, and input validation testing.
- His 1999 paper *Generating Tests from UML Specifications* created a new area in software testing, model-based testing, which now has several yearly conferences and thousands of papers. Model-based testing is used at thousands of software companies.
- With students, he developed several software programs that are used by thousands of software engineering researchers and educators, most notably Mothra, muJava, graph coverage web app, and logic testing web app.
- He has been invited to give keynote talks, invited lectures, and other presentations at conferences, universities, and companies throughout the world.

Throughout his career, he has invented numerous software engineering solutions that have been adopted by thousands of software companies worldwide. As a PhD student, while most software engineering researchers were describing ideas and claiming working software that was in reality "vapor-ware," his PhD advisor emphasized "*In my lab, we build software*." That is, if the name of the field includes "engineering," the research must be relevant to real engineers. He passes his advisor's principle on to his own students as "*our job is to find ways to help real software engineers build better software,*" and has adapted it as a tagline to the journal he edits as "*useful research in making better software.*"

We build software. DARPA funded two major software engineering research projects in the 1980s. One was for five years, 50 million dollars, across six universities and resulted in about a dozen publications, three or four PhD dissertations, and not a single line of working software. The other was for only three years and 5 million dollars to a small group at Georgia Tech. Georgia Tech's Mothra project resulted in more than 100 publications, almost a dozen PhD dissertations, and a large-scale research and educational software testing system that was used by hundreds of researchers and teachers throughout the world. As a PhD student, Offutt helped write the Mothra proposal, wrote more than half of the 100,000 lines of code in the Mothra software, and published dozens of papers about software testing

[1] The *h-index* measures impact of research through citation counts. The *h-index h* is the largest number *h* such that *h* publications have at least *h* citations. The index assesses the relevance and value of the scientist's most-cited papers and omits papers that were less relevant. Every field is different, but in software engineering having an h-index equal to the number of years of service is considered competent and having an h-index double the number of years of service is considered research star quality. Offutt graduated in 1988, so his h-index is just over two times his years of service.

as part of his PhD studies and after. The core paper from his dissertation, *Constraint-based automatic test data generation*, is listed on Google Scholar as having more than 1000 citations.

The Mothra project did more than revolutionize software testing. It set new standards for empirical research in software engineering. "*We build software*," unusual for researchers in the 1980s, is now such an established practice in software engineering that current scientists cannot imagine not building. Mothra was also the first research software system that was shared with source and documentation; as such, it has been called a prototype for modern open-source software (although in the 1980s it was distributed with floppy disks and envelopes).

Software testing is difficult, complicated, and incredibly important to the success of most modern companies. Software engineers use testing to find as many bugs as possible during development, trying to avoid inflicting them on users. Finding and fixing software bugs after the software is deployed costs 50 times as much as during development and the software failures they cause can, and often do, lead to significant loss of money, equipment, and lives. So better software testing saves money, improves safety, and increases user satisfaction. And increases sales.

Offutt has invented, developed, and experimentally validated numerous algorithms and engineering techniques in all aspects of software testing, and his inventions are used in thousands of companies in Virginia, the US, and world-wide. He is considered the premier worldwide researcher in mutation testing, which is widely acknowledged as the most effective software testing technique. It is also the most expensive. Offutt's research has automated key steps in mutation testing, established its value and relevance, and explored numerous ways to utilize it more efficiently and effectively. His current project in minimal mutation (collaboratively with Dr. Paul Ammann of George Mason, Bob Kurtz of Raytheon, and Dr. Marcio Delamaro of USP Brazil) has excited the community as having the potential to reduce the cost of mutation testing by one to two orders of magnitude, a result that can revolutionize software testing.

In the late 1990s, Offutt worked with engineers at Rockwell-Collins to develop methods to use their design descriptions to design tests to reduce the risk of software-controlled aircraft. As part of that project, he invented *model-based testing*, a technique that uses the design of software (roughly equivalent to an architectural blueprint) to design and generate tests for the software. His seminal 1999 paper on model-based testing, *Generating tests from UML specifications*, had a major impact on the software engineering field. Model-based testing is used by thousands of companies worldwide, and is a major focus of research, with dozens of papers published every year.

Offutt has also invented widely used and novel techniques to test unique aspects of web software. We all use web applications on a daily basis to purchase products, renew our driver's licenses, cross borders, read email, keep our calendar, and dozens of other tasks. Offutt showed how web software applications are different from other software at a detailed, technical level in his fundamental 2002 paper *Quality attributes of web software applications*. His atomic section model from his 2010 paper, *Modeling presentation layers of web applications for testing*, has been used by researchers to develop better ways to design, build, and test web software, and by software companies to create better quality web software. Offutt also invented bypass testing for web applications, which is used widely in the software industry to test web software for correctness as well as for security.

Offutt also made fundamental improvements to the way input space partition (ISP) testing is used in practical situations. ISP is used widely throughout the industry, and his invention of base-choice testing is by far the most common way ISP is applied. In collaboration with a full-time software test manager and part-time student at Freddie Mac, Offutt was able to show major benefits in effectiveness and significant cost savings when ISP was used. Their techniques are now used widely throughout Freddie Mac, one of the largest developers of software in Virginia.

His technical leadership and many successes as a scientist have led him to critical leadership roles in his professional community. For 11 years, he has been editor-in-chief of the premier research journal in his area, Wiley's journal of Software Testing, Verification and Reliability. He also co-founded the IEEE International Conference on Software Testing, Verification and Validation (ICST), was its first steering committee chair, and its program chair in 2009. His primary goal in these two research service activities has been to create a research community for young scientists that is rigorous, fair, supportive, and nurturing.

TEACHING. Professor Offutt has been teaching for almost 30 years, 26 at George Mason. In that time, he has:

- Been awarded Mason's Teaching Excellence Award, Teaching With Technology, in 2013. He was the first to be granted this award by unanimous vote.
  - He was awarded his department teaching award and the Volgenau School of Engineering teaching award multiple times
- Authored the textbook *Introduction to Software Testing*, with Paul Ammann, Cambridge University Press, second edition 2016 (with Dr. Paul Ammann).
- Received a 3-year, $900,000 grant from Google to re-design how introductory programming classes is taught (the SPARC project).
- Created 10 new courses, at all levels, and with all new material, several of which have been widely copied at other universities. Substantially revised more than a dozen other courses.
- Published more than a dozen academic papers on software engineering and computer science education.
- Has led and participated in efforts to create new degree programs and update existing programs. He is currently Director of the MS program in Software Engineering.
- Graduated 18 PhD students. Eleven were women, in a field where less than 8% of all PhDs are female.
- Mentored numerous students and junior faculty on all aspects of teaching.

Offutt teaches undergraduate, MS, and PhD classes. He was the first at George Mason to post course materials on his website and one of the earliest adopters of powerpoint in the classroom. He uses online discussion boards in every class to expand conversations beyond the classrooms. He has created nine courses at GMU, three at each level. Seven were new classes that had never been taught anywhere and were created without prior models or adequate textbook support. These courses fundamentally transformed Mason's software engineering programs, and are copied at universities around the world. Offutt created and first taught seven of the 14 software engineering courses offered in fall 2018, and the current instructors rely his materials. Many of these novel courses were either **derived from** his research, or **resulted in** new research projects.

Offutt's novel approaches to teaching have significantly influenced software engineering education at Mason and beyond, and are often copied internationally. Software engineering is by nature interdisciplinary, and his courses incorporate elements from computer science, engineering, management, and psychology. He eagerly embraces new teaching methods and regularly invents innovative techniques that continue to be adopted at universities throughout the world.

In his first year at Mason, Offutt redesigned SWE 637, Software Testing, introducing a new book and completely new material. This changed the emphasis in the course from research to relevant practical engineering that directly supports the needs of our MS students and companies where they work. With Dr. Paul Ammann, he re-invented the course again, this time with their own textbook, *Introduction to Software Testing* (Cambridge Press, 2nd edition 2016). The book is the most widely used software testing text in the world. The high quality slides, example assignments, solution manual, and support software make it easy for other faculty to use it. Offutt and Ammann donate all royalties to the Software Engineering Scholarship Fund that they created at Mason (over $30,000 so far).

Professor Offutt has been invited to teach software testing at universities and companies, including the ARTES summer school in Sweden, Samsung electronics in Korea, Rockwell-Collins in Iowa, Skövde University in Sweden, Ewha University in Korea, the TAROT summer school in Austria, and the Universidad Politécnica de Madrid in Spain.

Offutt also created a graduate course in designing and building software user interfaces that, when invented, was unique in looking at software usability as an engineering discipline (SWE 632, User Interface Design and Development). This course was long the most popular graduate electives in the Volgenau School, and was recently made a required course in the Master's program in Software Engineering on request from the department's industrial advisory board. It emphasizes collaboration, critical thinking, analytical thinking, and imaginative thinking in an interdisciplinary way. Elements from this course have been copied at universities throughout North America.

In 1999, Offutt created one of the first courses in the nation on engineering high quality Web software applications (SWE 642, Web Application Design and Development). The class was a direct response to the specific needs of the Northern Virginia software industry and has technical depth and immediate relevance. 642 is taught every semester, often by adjuncts using Offutt's materials. No textbooks adequately support this class (still!), so he designed it from "whole cloth." The class is innovative in both content and delivery, and it has been copied by dozens of universities. Offutt has taught this material in China, Sweden, and Austria at four universities and three companies.

Offutt also created and taught three undergraduate courses: SWE 205 Software Usability Analysis and Design, SWE 432 Design and Implementation of Software for the Web, and SWE 437 Software Testing and Maintenance. SWE 432 is one of the most popular electives for CS majors and the department offers two sections when possible. SWE 205 is the best loved, with an average student evaluation rating of 4.94 out of 5.00.

Offutt also created several research-related courses for PhD students, including SWE 763 Software Engineering Experimentation, IT 821 Object-Oriented and Architecture-based Testing, IT 824 Analysis of Software for Testing, and SWE 825 Special Topics in Web-Based Software.

Talk less, teach more. These successes in the classroom helped Offutt win George Mason University's highest teaching award, the Teaching Excellence Award, Teaching With Technology, in 2013. Since then, Offutt has dedicated even more energy to improving teaching results. He redesigned most of his classes to emphasize in-class work and collaboration, while reducing time spent lecturing. This is based on evidence that most students learn far more from doing than from listening. Thus, by talking less, he is teaching more. In 2013, he was an excellent lecturer. Now he is a world-class teacher.

Teaching computer science and software engineering students at a Virginia university is a privilege, made better by many excellent colleagues. Mason's undergraduate software engineering students are serious, intelligent, and hard working, partially motivated by the knowledge that they live in a place with a hot economy and their skills and knowledge are in demand. Many are first generation college students or have immigrant parents. This melting pot of a teaching environment adds excitement, increases the opportunities for students and faculty to learn, and adds interesting challenges. At the graduate level, more than half of the students work part-time in the local software industry. This means that classes must be both relevant and current. More than a third of the graduate students are foreign-born. All of these factors make the job exciting and challenging. For two and a half decades, Offutt has had great success using his passion and the excitement this diversity brings to rise to the challenge, teaching thousands of students who are in leadership positions in the software industry throughout the region, state, nation, and world.

INTEGRATION OF KNOWLEDGE. Offutt has never separated research discovery, teaching, service, and industry relevance. When the software industry shifted to object-oriented (OO) software in the 1990s, he integrated this industry shift into his teaching by re-designing SWE 619 Software Construction to emphasize OO concepts, and then into his research by winning an NSF grant to invent new ways to test OO software. The results are featured in his textbook. When web software became important in the late 1990s, he created two new courses (SWE 642 and SWE 432) to prepare Mason students for this important industry trend, and initiated several new research projects in Web software. He now has approximately 20 refereed publications on aspects of testing and modeling web software, and his inventions are used widely in the software industry.

Service to improve education: George Mason's MS in Software Engineering degree (MS-SWE) was created in 1989 and is one of the oldest and most successful in the nation. Offutt has taught 11 different software engineering classes and served as Director of the MS-SWE program from 2003 to 2009, and again since 2016. He led the creation of a concentration in Software Engineering within the PhD in Information Technology program (2000), an undergraduate concentration in Software Engineering within the Applied Computer Science program (2010), a PhD program in Computer Science (1999), and helped with a major revision in 2017. He also led major renovations to the MS-SWE degree in 2005, 2010, and 2018. The changes were both innovative and fundamental, and both students and the department's industrial advisory board have been overwhelmingly in favor of the curricular modernizations.

SWE 763: Teaching one class at three universities: Offutt's most innovative class started with an unusual question: *How can one professor teach the same class at three different universities?* It resulted in novel research in educational methodologies and multiple student research projects that were published in peer-reviewed outlets.

Offutt first created SWE 763, Software Engineering Experimentation in the 1990s. It was a traditional, face-to-face class that met once a week for 2.5 hours. The class featured three weeks of lectures about designing, conducting, and reporting on experiments, nine weeks of in-class discussions of case study research papers, and two weeks of students presenting their project results. In 2011, a collaborator in Sweden (Birgitta Lindström) asked Offutt to teach 763 at Skövde University in Sweden. That was impossible, but the request led to the above question.

Teaching a class at multiple, international, universities presents numerous problems. Lectures could not be synchronous because of time zones. Mason regulations prohibited Offutt from being the "instructor of record" at another university while teaching at Mason. Would the students be enrolled at GMU or their home institutions? USA students pay tuition directly to universities but tuition is paid by tax money in Sweden. Lastly, whose learning management system (LMS) should be used? It is very difficult to enroll students from one university into another university's LMS, and the poor usability of the discussion board commonly used at Mason would not support the kind of dynamic interaction needed.

The solution came in several parts. Mason's Provost gave special permission for Offutt to be instructor for the classes in Sweden, allowing students to enroll at their home universities. The second part was an accidental spam message from a young startup company, piazza.com. Their tool is a free discussion board with a modern, social networking style user interface—exactly what was needed!

The distributed, asynchronous, online version of SWE 763 had the same structure as in 2008—three weeks of recorded lectures followed by online discussions, nine weeks of online discussions, and two weeks of student presentations. The difference was it was entirely online and asynchronous. Once the class was announced, Linköping University (also in Sweden) asked permission for their students to join, making three.

In spring 2012 the course had 20 students; 14 from Mason, four from Linköping, and two from Skövde. The students quickly melded into one unified class, answering the question affirmatively: Yes, a professor could teach the same class at three different universities. The introductions were more detailed and informative than in face-to-face classes, which foreshadowed the major benefit of the asynchronous format.

The discussions were nothing less than **outstanding**. The traditional classroom meetings had a 2.5 hour per week limit, students often came to class tired, many after a long day at work, and not always prepared to discuss the papers thoroughly. The asynchronous format allowed students to extend the discussions indefinitely, prepare at their convenience, consider other student's comments, and add to the discussion anytime. The discussions were both longer and deeper than in the face-to-face class! The online discussions were fascinating and the students taught the professor and each other.

The advantages were highlighted during the week of project proposals. In 2008, the students spent five minutes apiece presenting their experimental designs, and the professor spent five minutes giving them feedback. In 2012, students posted 10 to 15 minutes worth of material online, and received up to **four hours**' worth of feedback. Some discussions continued for several days. As a result, their projects were significantly better in 2012. The online format freed the class from the "tyranny of the clock"!

The online format also led to significant integration between the educational and research aspects of the class. In 2008, four of 15 papers (27%) were eventually published. In 2012, ten of 20 (50%) student papers were published. The format was reused in 2015, this time including a university in Brazil, with similar success, and is being taught again in Spring 2019.

This format can be successfully copied with any discussion class, whether in scientific research or in literature. The discussions and diversity of projects meant the class emphasized divergent thinking, reducing the problem of determining who actually did the work, a common problem in online classes. Moreover, the multi-university format offers several interesting possibilities. If there aren't enough students to support a specialized class at one school, they could join the same class at another university, even on another continent. If a class is taught by someone with unique expertise, students elsewhere could join. This format could scale from 20 to 200 students if professors at other universities participated in the feedback and evaluations. This opens the possibility of "crowd teaching," where a diverse team of professors from 5 or 50 universities merge their courses online, all being responsible for part of the content, but each professor being responsible for managing a subset of the students.

The SPARC Project: Since 2015, Offutt has led a multi-disciplinary team of eight faculty members in a 3-year, $900,000 educational grant from Google to try new techniques to teach introductory programming. The key ideas of the SPARC project are to separate practice programming from graded assessments, allow self-pacing, encourage lots of collaboration and peer instruction, provide multiple chances at graded work, intervene with struggling students, and take advantage of online learning. The primary goals were to increase the rate at which students passed and to reduce the cost of teaching. Both goals were reached. In addition, students caught cheating went from around 20% to zero, and students and teachers alike reported much more satisfaction with the courses.

PhD advising: Offutt has also been enormously successful at integrating research and teaching through his prolific mentorship of PhD students. He has advised and co-advised 18 PhD students. Six are university professors, 10 are technical and executive leaders in industry, and two are consultants. Perhaps most impressive is that despite working in a field where less than 8% of all PhDs are female, 11 of his 18 PhD advisees are female.

SERVICE TO THE INSTITUTION, COMMUNITY, AND PROFESSION. Service is part of every professor's job, but usually only a small part. At Mason, 20% of a professor's time is officially allocated to service, although service is given zero weight in promotion decisions. Offutt has gone far above this 20% minimum in many ways. He gives hundreds of hours every year to committee and administrative work, mentoring junior faculty, advising students, writing references, reviewing papers, editing journals, organizing conferences, giving invited talks, and reaching out to the local community as a speaker, coach, and youth baseball umpire.

- **Computer Science Department**: Offutt was department chair (1999) and has been a member or chair of virtually every departmental committee. He designed and built the ISE department's first website, and currently is Director of the MS program in Software Engineering and Chair of the Graduate Studies Committee.
- **Volgenau School of Engineering**: He currently serves on VSE's graduate studies committee and online learning committee. He has past served on Dean's hiring and reappointment committees, VSE's P&T committee, and VSE's grievance committee.
- **George Mason University**: He has served on the Provost's reappointment committee, various search committees, excellence in teaching awards committees, and wrote the first version of Mason's Responsible Use of Computing policy.
- **Research Community**: He is editor-in-chief of the premier journal in his field, *Software Testing, Verification, and Reliability*. He co-founded and was the first steering committee chair of the top conference in his field, IEEE's *International Conference on Software Testing, Validation, and Verification*. He was program chair of several conferences, as well as chair of PhD symposia, workshops, and various other academic meetings. He has reviewed several thousand journal and conference papers in his career. He writes more than a dozen letters of recommendation for faculty being considered for promotion every year.

Local Community: Offutt enjoys reaching out to the local community and does so often. He gives talks through Mason's Speakers Bureau and visits local software companies to talk about current trends in software engineering. Most other local community activities are through Offutt's three children. He has volunteered dozens of times at local schools (Oak View ES, Robinson SS, Thomas Jefferson HS), coached his son's Odyssey Of the Mind team for four years, and coached youth sports for more than 20 years. He volunteers as a little league umpire 20 or 30 times per year.

SUMMARY: Part of Mason's mission is to "*help our community thrive culturally, socially, and economically.*" Through new degree programs, new classes, innovative teaching, multiple inventions, and hundreds of hours speaking at software companies, Offutt has helped this community thrive in major ways. Knowledge is essential to the success of individuals and the growth of humanity. Offutt studies to acquire knowledge, he carries out research to increase our knowledge, and he teaches to share that knowledge with others. Every person should have the opportunity to learn as much knowledge as they want and can handle, and his passion for learning leads directly to a passion for creating new knowledge and giving that knowledge to anyone who wants it. As a conversationalist, he can be boring and pedantic by constantly either asking for new knowledge or sharing his knowledge. As a teacher, his students are fortunate to have the opportunity to learn from this talented and hard-working professor.

Section D. Personal Statement of Professor Jeff Offutt

I love teaching—I love organizing the material, lecturing, helping students, grading papers, and everything else. My favorite part is when a student struggles with a problem or a concept, and I diagnose the difficulty, explain the concept, and see the student get the solution. This happens in every subject, every level, and with every student. Seeing that particular light bulb ignite, up close and personal, still thrills me.

My educational background. More than teaching, I love learning. I grew up in the Appalachian educational desert. Teachers taught to the slowest students, and classes were torture for fast learners. I was bullied for being good at math. Skipping fifth grade increased the bullying, and when the school placed four special-ed 15 year olds into my small 8th grade class, all I learned was how to take a punch. 40% of my high school classmates dropped out, and only 10 of 150 graduates started college. My geometry teacher couldn't do proofs, my chemistry class had no chemicals, and English classes included functionally illiterate students. By the end of high school, I was poorly educated, but hungry. Starved for knowledge.

I went to a local university because it was what I could afford on my own, working part-time jobs in fast-food restaurants. After two years of being a desultory college student, I realized I was lazy and knew nothing about studying. So I started to learn how to study and how to work hard. I was fortunate to land a summer internship at Exxon Research Labs and learned three important things. (1) Research was fun! (2) The other interns from schools like MIT, Stanford, Berkeley, and Harvard knew a lot more than I did. (3) They weren't inherently smarter.

I was then lucky to get into a great graduate school. The department made an argument for more faculty by dramatically increasing the number of graduate students. The professors weren't all on board and some tried to reduce class size by being very demanding. I knew I couldn't know more than my fellow students, but I could choose to work harder. I had good health and lots of energy, and used both to study 7 days a week, 16 hours a day. And found great teachers to feed me.

After 35 years of studying, I'm still hungry. I absorb knowledge as enthusiastically as my 16-year old son absorbs pizza.

Parenting and teaching. When my daughter was born, I realized I never wanted her to suffer that hunger. And I wanted her to be taught math the same way boys were, not to be told "*it's okay honey, girls don't need to be good at math*." So I looked for a place with great K-12 schools. In 1992 I had 5 options and I chose FCPS for my daughter and George Mason University for potential, innovation, and diversity. She affirmed my decision by becoming an engineer at Virginia Tech and Mason affirmed my decision by being an ideal place for me to grow as a teacher, a scientist, and a person.

I think in abstractions, and teaching, research, parenting, and coaching are all the same to me. That is, I view parenting as teaching and teaching as parenting. My children taught me how to teach and my students taught me how to parent. When my daughters were named captains of their high school Color Guards, we developed a list of leadership rules. I quickly adapted them as teaching principles:

1. If you don't care about them, they won't care what you say
2. Don't expect them to be students like you were
3. The main correlation with success is frequent detailed feedback—not class size, difficulty, quality of lectures, or the book
4. Respect them as people, even if they don't earn it with results
5. Fear is not respect
6. Teach to individuals, but grade anonymously
7. Don't apologize for doing the right thing, but always apologize for mistakes
8. If you don't know the answer, help them find it
9. If they think you want them to learn, they will learn more

These principles have guided me for years and were crucial to being awarded Mason's Teaching Excellence Award, Teaching With Technology, in 2013. Ironically, that only started me on a new journey. Because of that award, I started attending Mason's Innovations in Teaching and Learning conference, where I learned I was doing it wrong. After 25 years of trying to become a "great" lecturer, I learned that students learn more from active classrooms and peer

learning. My current teaching motto is "***Talk less, teach more***," and my students now learn more by doing than they ever did by listening.

Parenting created a desire to pass my knowledge on. That translated to teaching and to research.

Teaching and research.

I view research is another form of teaching, where I am both teacher and student. I love to be the teacher I seldom had, and teaching is a way to cement that knowledge in my head and pass it on to others. Teaching, research, and parenting are all the same to me. The Engineering school at Mason (VSE) requires professors to be excellent in research or teaching to be promoted. And we all know that in VSE, that really means research. I am especially proud to have been the first professor in VSE to be promoted for excellence in **both** research and teaching.

I am proud to have published more than 180 fully-refereed research papers. I am proud to have the highest impact factor in my department (61 on Google Scholar). I am proud that my ideas and innovations are used by other scientists and widely in the software industry. But I am most proud for mentoring 18 successful PhD students. 11 are female in a field where more than 92% of all PhDs are male. They all have successful careers as corporate executives, professors, senior software engineers, and consultants.

Service is teaching plus research plus parenting.

My passion for all of my service activities comes from helping young people avoid the hardships I faced. In my dozens of committees and administrative roles at Mason, I have fought to improve our recruiting and hiring processes, to make it easier for new faculty to start and succeed, to be fairer to students, to improve communication, and to reduce friction in the department.

When I started, my field of software testing was plagued by bad science, poor validation techniques, and publication decisions that were based more on people than on scientific merit. I wanted to make my field more congenial and fairer for young scientists, so I became editor of our primary journal, created a new IEEE conference, and wrote the most widely-used textbook. Every year at the conference, young scientists thank me and my co-founders for gifting them with a place where they respect the decisions and can see examples of top-notch science.

I also try to give back to our local community. I have spoken at dozens of local clubs and organizations through Mason Speakers Bureau; visited numerous public schools to talk about computing, college, and learning; and coached and officiated youth soccer, odyssey of the mind, baseball, and basketball. I view all of my youth service as preparing kids to become responsible and successful adults. I coached my son's odyssey of the mind team four years, where we learned divergent thinking, creative problem solving, teamwork, and independence. I view baseball is a conduit to teach resilience, the value of hard work, how to balance cooperation and competition, and how to combine fitness with thinking. Many of the kids I coached are now on their high school baseball teams, and more importantly, all of them are getting good grades.

Who I am. I don't just have a job teaching, I am fundamentally a teacher. Sometimes it annoys my teenage and adult children, but I can't help it. I am a scientist. I am a parent. I am a conduit for knowledge. I was never satisfied with the opportunities I had as a kid, and my goal in my profession and life is to help as many people as I can have as many opportunities as they can handle. This is true for my colleagues, young scientists in my field, my students, and all the kids I can reach. And, selfishly, I am still a learner. I learn a little slower than I did 35 years ago, but I'm still hungry.

Section E. Abbreviated Curriculum Vitae of Professor Jeff Offutt

Education

- PhD in Computer Science, Georgia Institute of Technology, 1988
- MS in Computer Science, Georgia Institute of Technology, 1985
- BS in Math and Data Processing (double major), Morehead State University, 1982

Academic Appointments

- Full Professor of Software Engineering, George Mason University, 2005–
- Part-time visiting professor, Department of Computer Science, University of Skövde, Skövde Sweden, 2002–
- Associate Professor of Information & Software Engineering, George Mason University, 1996–2005
- Department Chair, Information & Software Engineering, George Mason University, 1998–1999
- Assistant Professor of Information & Software Engineering, George Mason University, 1992–1996
- Assistant Professor of Computer Science, Clemson University, 1988–1992

Selected Scholarly Works and Activities (*187 peer-reviewed papers; over 17,000 citations; student names bold*)

- Paul Ammann and Jeff Offutt, Introduction to Software Testing, second edition, 2016, Cambridge University Press (first edition 2008)
- Jeff Offutt and **Sunitha Thummala**. Testing Concurrent User Behavior of Synchronous Web Applications with Petri Nets. Springer's Software and Systems Modeling, 2018
- **Nan Li** and Jeff Offutt. Test Oracle Strategies for Model-based Testing. IEEE Transactions on Software Engineering, 43(4):372-395, 2017
- Deanna D. Caputo, Shari Lawrence Pfleeger, M. Angela Sasse, Paul Ammann, Jeff Offutt, and **Lin Deng**. Barriers to Usable Security? Three Organizational Case Studies. IEEE Security & Privacy, 14(5):22-32, 2016
- **Upsorn Praphamontripong**, Jeff Offutt, **Lin Deng**, and **JingJing Gu**. An Experimental Evaluation of Web Mutation Operators. IEEE Workshop on Mutation Analysis, 2016
- Mark Ardis, David Budgen, Gregory W. Hislop, Jeff Offutt, Mark Sebern, and Willem Visser. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer, 48(11):106-109, 2015
- Paul Ammann, Marcio E. Delamaro, and Jeff Offutt. Establishing Theoretical Minimal Sets of Mutants. IEEE International Conference on Software Testing, Verification, and Validation, 2014
- Jeff Offutt and **Chandra Alluri**. An Industrial Study of Applying Input Space Partitioning to Test Financial Calculation Engines. Springer's Empirical Software Engineering journal, 19(3):558-581, 2014
- Jeff Offutt, **Vasileios Papadimitriou**, and **Upsorn Praphamontripong**. A Case Study on Bypass Testing of Web Applications. Springer's Empirical Software Engineering journal, 19(1):69-104, 2014
- Jeff Offutt. Putting the Engineering into Software Engineering Education. IEEE Software, 30(1):96-100, 2013
- **Roger T. Alexander**, Jeff Offutt, and Andreas Stefik. Testing Coupling Relationships in Object-Oriented Programs. Software Testing, Verification, and Reliability, 20(4):291-327, 2010
- **Jane Hayes** and Jeff Offutt. Recognizing Authors: An Examination of the Consistent Programmer Hypothesis. Software Testing, Verification, and Reliability, 20(4):329-356, 2010
- Jeff Offutt and Ye Wu. Modeling Presentation Layers of Web Applications for Testing. Software and Systems Modeling, 9(2):257-280, 2010
- **Mats Grindal**, **Birgitta Lindström**, Jeff Offutt, and Sten F. Andler. An Evaluation of Combination Testing Strategies. Empirical Software Engineering, 11(4):583-611, 2006
- **Jane Hayes** and Jeff Offutt. Input Validation Analysis and Testing. Empirical Software Engineering, 11(4):493-522, 2006
- Jeff Offutt. Quality Attributes of Web Software Applications. IEEE Software, 19(2):25–32, 2002
- Jeff Offutt, **Zhenyi Jin**, and **Jie Pan**. The Dynamic Domain Reduction Procedure for Test Data Generation. Software Practice & Experience, 29(2):167–193, 1999
- Jeff Offutt and **Aynur Abdurazik**. Generating Tests from UML Specifications. International Conference on the Unified Modeling Language, 416–429, 1999

- Jeff Offutt and **Stephen D. Lee**. An Empirical Evaluation of Weak Mutation. IEEE Transactions on Software Engineering, 20(5):337–344, 1994
- Jeff Offutt. Investigations of the Software Testing Coupling Effect. ACM Transactions on Software Engineering Methodology, 1(1):3–18, 1992
- Rich DeMillo and Jeff Offutt. Constraint-Based Automatic Test Data Generation, IEEE Transactions on Software Engineering, 17(9):900-910, 1991 (The main result from Offutt's PhD dissertation.)

Recent PhD Students (*of 18 total PhD students, plus 15 MS theses advised*)

- Lin Deng, August 2017, Assistant Professor, Towson University
- Upsorn Praphamontripong, May 2017, Assistant Professor, University of Virginia
- Jing Guan, May 2015, Senior Software Engineer, Lockheed Martin
- Nan Li, June 2014, Senior Software Engineer, Medidata Solutions
- Jing Jin, December 2013, Senior Software Engineer of Security @ Intuit
- Gary Kaminski, 2010 (co-advised with Dr. Paul Ammann), Software Engineer at CACI

Inventions, Innovations, and Software (*collaborators are omitted for brevity*)

- Offutt has contributed dozens of major results to **mutation analysis**, collectively resulting in a comprehensive engineering solution that makes mutation practical for industrial use
- Invented the first technique for **model-based testing**
- Difficult technical solutions for a**utomatic test data generation** are used in commercial tools Agitator and Pex
- Developed new methods and processes to apply **input space partitioning**
- Developed metrics to evaluate software maintainability, **OO software**, **coupling**, and **component-based software**
- Invented several techniques to test web applications (**data perturbation, bypass testing**, **finite state machine modeling**, and **atomic section modeling**)
- Invented **prime path testing**, **base choice testing**, **coupling-based testing**, **input validation testing**

Selected Funded Projects (*out of over $4.5 million in external funding; collaborator's names omitted for brevity*)

- Preparing K-5 Teachers to Integrate the Computer Science Standards of Learning in Inclusive Classrooms to Support Students with High Incidence Disabilities, PI Amy Hutchison, NSF CS for all, $999,423
- SPARC: Self-PAced Learning increases Retention and Capacity, Google Education Grant, $900,000, 2015-2018
- Usable Analysis of Security Protocols, Department of Homeland Security, $128,993, 2012-2014
- Research into Testing Service Oriented Architectures, Avaya Research Labs, $61,169, 2006-2007
- Assuring Web-based Software System Components, NASA Goddard, $104,331, 2003-2004
- Repeated Maintenance of Open-Source Software, NSF: CCR-00-97056, $225,000, 2001-2004
- Coupling-based Analysis for Integration Testing of OO Software, NSF: CCR-98-04111, $200,000, 1998-2001
- A Comparative Evaluation of Data Flow and Mutation Testing, NSF: CCR-93-11967, $107,390, 1993-1996
- Generating Test Cases From Requirements-Specifications, Rockwell-Collins Avionics, $125,000, 1997-2000

Honors and Awards

- George Mason University Teaching Excellence Award, Teaching With Technology, 2013
- Finalist, Governor's Technology in Education Award, 2012
- GMU Outstanding Faculty member, 2009, 2010
- Putting the Engineering into Software Engineering Education, IEEE Software, was selected by ACM Computing Reviews as a notable article for 2013
- Outstanding Teacher Award, ISE Department, 2003, School of IT&E, 2003
- Outstanding Paper Awards, Mutation Workshop 2018, ICECCS 1996
- Outstanding Researcher Award, School of IT&E, 2004, ISE Department, 2006

Other Professional Activities

- Organized multiple international conferences and workshop
- Consulted to train software engineers, establish best practices, and help solve relevant practical problems
- Expert witness on various legal cases, mostly involving software intellectual property

Section F. Excerpted letters of support for Professor Jeff Offutt

(*Names deleted for public posting*)

Academic peers

"Dr. Offutt is an internationally leading researcher in the field of software testing. Offutt's work is both foundational and seminal, and has influenced software engineers around the globe, myself included. I am sure that the impact of this work will be felt for considerable time to come. Offutt is a world-class and leading scientist in an important and highly competitive field of research.

"One thing that always impressed me is Dr. Offutt's capacity to get good, practical ideas, and to present them clearly to the reader. I teach many of Dr. Offutt's research works in my graduate courses. I find his research relevant, clearly presented, and of practical importance. His attention to solving important, practical problems gives his research a much stronger impact than average academic contributions. He is a stellar asset to GMU."

"Jeff is one of the most important software engineering researchers working today. His work on mutation testing now spans four decades and Jeff has been the most prolific contributor in the world to the literature. It's work that I like for its elegant engineering approach. The most striking thing about Jeff's career is the extent to which he has managed to weave an impressive research career with a genuine love of teaching. His students are always motivated and prepared—a real accomplishment considering the high standards he sets for himself in the research domain."

"When the International Symposium on Software Reliability Engineering decided to include software testing, Offutt was the obvious and first choice to join the PC. His contributions are numerous and include basic theoretical results, systems/development work, and substantial experimental results. Many of his papers are read quite widely and referenced continuously. When Jenny Li from Avaya Labs asked a student to study automatic test data generation, the student came back and said "Offutt has already done it all." I'm also struck by his very broad range of collaboration. I have seen Offutt give several talks at conferences, and always been impressed by his organization and ability to make complex ideas seem simple. I have also seen talks by his former PhD students and found them to be almost equally impressive. I have more than once been struck by how the "younger generation" of scientists and students respect his work. They have been reading his papers as seminal work in software testing and are also impressed that he continues to produce great results. Being highly respected by our peers is wonderful, but the respect from our junior colleagues may be more meaningful because it reflects our potential for lasting impact."

"Offutt's work in object oriented development has been remarkably broad, ranging from integration testing, to quantitative analysis of various aspects of OO programs testing UML designs. He really was ahead of the curve when he started publishing in this area. Two former PhD students, Dr. Alexander and Dr. Hayes are glowing examples of Dr. Offutt's effectiveness. Offutt has highly innovative ideas and is immense fun to do research with. He wins students' respect through the respect he shows them. He has a deep knowledge of the subject and very effectively communicates it. I am particularly impressed by his innovative approach to new technology."

"I performed yearly evaluations of our faculty, split along three lines, research, teaching, and service. Offutt was seldom ranked 1st in any category or for any one year, but was always top 4 in every category. Over any 3-year window and averaged over all 3 categories, he always came out #1."

Industry leaders

"I lead a team of over 100 engineers that builds software for the federal government. More than 15 of Offutt's former students work with my project. They are managers, team leads, senior software engineers, testers, and junior developers, and they all share two things in common. First, they are all excellent team members. Second, they demonstrate every day how much they learned from George Mason in general and Offutt in particular. They understand the theory behind software testing and software development, but also know how to use it in real life. We are an agile project that emphasizes unit testing and test automation, and we have won multiple awards from the US Government. I give credit to the hard work from my team members and their ability to apply what they learned from Professor Offutt. The software we deliver was built by Pyramid, but you could say that Pyramid was built by Offutt."

PhD students

"His classes were some of the best that I experienced at George Mason University and led directly to insights that have assisted me in my career. I am required to develop novel approaches to unsolved information security challenges. I can point to ideas that Jeff created that I use daily. One of the themes I use in information security is that unmanaged complexity leads invariably to insecurity. This concept that was first driven home to me in the first class I took from Dr. Offutt on the topic of Software Testing. This class provided tremendous insight into the reasons faults get into software and how to find and fix them. Dr. Offutt did not focus the class purely on theory. He explained the why of the subject but also helped me to learn specific real-world testing techniques. I noted an increased ability over time to take the insights he developed in his research and apply them to the learning environment. Because of this, his classes deliver concepts that are at the state of the art for their subject areas. Dr. Offutt has a true passion for his field and a life-long interest in teaching. This convergence makes him engaged in his classes with material and insight that propels students such as myself to approach the field and its challenges in new ways."

"Jeff Offutt is the best teacher I have ever had. Period. Not only did I learn about information technology and software engineering and computer science in his classroom, I learned what it means to be a professional, what it means to be a scientist, and what it means to take skills and education out into the world and make a difference. As I prepared to defend my dissertation, I was a Corporate Vice President and had 230 employees in six offices across the country. I was offered the Group Vice President promotion just one year after I received my PhD from GMU. Due to Jeff's influence on me, I decided to forgo the promotion and join academia. I still use his slides, with permission, in two of my courses (Software Testing and Empirical Software Engineering). I use his software testing textbook, by far the best textbook I use in any of my courses. I recently won the Hanover College Distinguished Alumni award. Jeff was the first non-family member that I thanked in my acceptance speech."

"Dr. Offutt not only gave me academic guidance, taught me many Software Engineering knowledge and techniques, but also taught me many lessons on managing career and life. He talked to me about time manage strategies on how to have the discipline to spent more time on "important" sectors of the time management quadrant so I can spent less time on "urgent but not important" and "not urgent/not important" sectors. He encouraged me to be persistent and stick it through during my long, arduous study journey. He is the best advisors and mentors I have met in my career. His encouragement and his wise advice on life, career, research and study have benefited me again and again in my life and career later on. His casual saying such as "you can never be burdened by having too much knowledge or regret you have learned too much" have encouraged me to grab every opportunity I can get to learning, including an MBA six years after the PhD degree."

"Prof. Offutt's teaching in a vivid and effective way in his class has been motivating students to new discoveries and develop new applications. While I was working at VMware Inc, I was able to apply what I've learned from his testing class to my daily work. I used his mutation testing technique from his testing class to test a product at VMware. Offutt stepped up to support my PhD study when I almost gave up. Offutt has been like a father to me because he taught me to never give up, never be afraid of challenge the authority, and develop my own views in research and career."

"I feel fortunate to have worked with Dr. Offutt during my PhD studies abroad. He was instrumental in my efforts to obtain my PhD degree. He is a gifted lecturer, his ability to impart knowledge to students is uncanny. The highlight of my work with Offutt was his Software Engineering Experimentation course. The course was taught concurrently at GMU and two universities in Sweden. Collaborating with people who have different background was by far a better learning experience. He diligently steered the discussions towards critical thinking."

"I first knew Jeff Offutt in 2002 when he became my PhD advisor. Jeff is an excellent researcher, he is patient and listens carefully and gives helpful advices. He realizes when a student needs detailed technical advice and when the student needs more general guidance. With Jeff as my advisor I grew both as a person and as a researcher. Today I have my own PhD student that Jeff is co-advising. He still supports me and I still grow. We have had many discussions regarding education and I have learned most I know about teaching from him. I have been using his teaching advices and

performed better as a teacher each year. In 2012, I got a best teacher award from my university. I strongly believe that I owe this to Jeff. As a teacher, Jeff is very creative and keen on new ideas and solutions for education. During 2012, we had a collaboration between George Mason, the University of Skövde and Linköping University. We gave the same distributed course in software experimentation PhD students in US and Sweden got together in on-line lectures and seminars and it was amazing to see all the activity between the students. There's no doubt that these students were learning a lot and enjoyed doing it. Moreover, there's no doubt that both Jeff and I learned a lot and enjoyed it."

"Professor Offutt was the most inspiring teacher in my academic and professional life. He recognized my strengths and taught me that any work can become great when you believe what you're doing is important. He supported me to conferences to interact with other researchers, in particular with female researchers, which I found motivational. He taught me the importance of collaboration, dissemination of knowledge, and clarity of communication. I have found the skills I learned from him while I was working with him invaluable in my professional life."

BS and MS students

"After I performed unusually poorly on two consecutive weekly quizzes, Dr. Offutt noted his concern on my returned quiz. A simple "You can do better" pushed me to make time to study for the next one, and he noted the recovery as well. Offutt prioritizes student engagement and requires students to use concepts in practice, not regurgitate information."

"Offutt cares about his students, and cared that we learn. He is the best teacher I have ever known."

"Offutt is the most innovative, diligent, knowledgeable and fun professor I have ever met. What makes Dr. Offutt an extrodanary professor is that he is a true innovator. The most amazing class I took is the Software Engineering Experimentation class using asynchronous technology to virtually merge GMU with two Sweden Universities. He inspires me to learn more, try more and think more. My proudest moment was when he gave me an outstanding student award at graduation. Imagine that—a new musician getting an award from a rock star!"

"His unassuming personality and jokes made students feel welcome and positive about interacting during class. How many professors show a slide listing their responsibilities to students? Offutt does. All profits from his textbook goes to a GMU scholarship fund. Does that resolve any potential conflicts-of-interest, making students respect and trust him? You bet."

"I took every course possible with Professor Offutt. The reason is simple—I had a lot fun. The fun led to engagement, the engagement led to learning and ultimately led to doing."

"When I considered moving from BUAA in China, my professors said '*go to George Mason, that's where Offutt is.*' I saw his character when I made the difficult decision to quit my PhD studies. Instead of being angry, Jeff simply accepted my decision and helped me find a job."

Section G. Additional documentation for Professor Jeff Offutt

Evidence of impact.

Google scholar[2] maintains web pages for prominent scientists that lists papers and how many times each paper has been cited as a rough measure of research relevance. Offutt's Google scholar page says over 17,000 papers have cited his work. This is in the top five in the world among software testing researchers. The *h-index* is the largest number, ***h***, such that ***h*** papers have at least h citations. As of August 2018, Offutt's h-index is 61. In his field of software engineering, it is common to view an h-index in relation to the number of years since the PhD was awarded. An h-index of half the number of years is considered minimally competent, an h-index equal to the number of years is considered excellent, and an h-index equal to twice the number of years is considered "star" quality. Offutt received his PhD 30 years ago, so an h-index of 61 demonstrates him to be a world-renowned and "star" scientist. This is also the highest h-index in his department and in the top 10 at George Mason.

Offutt's most cited publication is his textbook, *Introduction to software testing* (2016). By integrating classroom material with cutting-edge research material (his and others), this book is the first place young scientists go to learn the field. His next five most cited publications were all research papers that were published in major international venues and that have had enormous impact on the field of software testing. *Constraint-based automatic test data generation* (1991), with almost 1000 citations, presented the major results from his PhD dissertation. After almost 30 years, it is considered a seminal paper in automated test generation, and is still cited by dozens of papers every year. Not only has his paper *MuJava: An automated class mutation system* (2005) had well over 600 citations, it represents a dual purpose. The paper extended mutation research into object-oriented language features and also presented a major research tool that is used by hundreds of scientists an educators around the world. It is an open-source tool with a highly trafficked website that Offutt maintains[3]. His next most cited paper, *An experimental determination of sufficient mutant operators* (1996), showed how mutation testing could become an order of magnitude cheaper. This is a seminal result that in the past five years has led to an entirely new research thrust in mutation testing that appears to be revolutionizing mutation testing. Offutt introduced the idea of model-based testing in his seminal paper *Generating tests from UML specifications* (1999), as part of an industry-sponsored research project. Model-based testing is now used by numerous software companies, has multiple conferences and workshops, and is the subject of dozens of research papers every year. Another seminal paper, *Quality attributes of web software applications* (2001), demonstrated that the advent of web software completely changed the economics and the engineering of software. Instead of optimizing time-to-market and time-of-execution, web software engineers can optimize traditional engineering criteria such as quality of service, reliability, and usability. This paper is considered seminal in establishing the widespread relevance of all software engineering research related to the web.

Excerpts of public reviews of his book (Amazon, Goodread, & Cambridge University Press)

"This is a great testing book, both from academic and industrial perspectives. I believe Ammann and Offutt's book will become the testing textbook of choice. There are a lot of testing books out there, some better than others. Most are narrow in the topics they cover and the level of detail they present. In stark contrast, Ammann and Offutt's book has the advantage of presenting concepts and techniques that cover the broad range of languages and platforms used in practice by industry and academia. Theirs is one of the most thorough and practical testing books ever published."

"… well written, logically organized, and provides excellent examples as well as useful course materials (such as homework assignments and quizzes) … greatly facilitates the teaching and learning process."

"This software testing textbook creatively uses only four models to capture current software testing techniques helping students, researchers, and practitioners to deeply grasp these techniques from an abstract, systematic perspective. The textbook provides excellent balance between theory and practice!"

---

[2] Offutt's Google scholar page: https://scholar.google.com/citations?hl=en&user=fAeRp3kAAAAJ

[3] https://cs.gmu.edu/~offutt/mujava

"This is a great book for learning software testing. The clear terminology definitions and comprehensive examples provide an easy way to master the software testing techniques … for people who work in or [are] prepar[ing] to enter the software testing field, this book is definitely what you need in your hand."

"Where has this book been all these years? [It] reminds me why I wanted to study software testing in graduate school: testing touches on all the interesting theoretical/engineering models/representations for software."

"These two leading researchers provide an excellent exposition of the recent advancement of software testing in a manner that is suitable for classroom use. Theoretical concepts are covered rigorously and in practical contexts allowing students to build a solid foundation while being well-connected to the real applications. The abundant examples and exercises make both teaching and learning a more tangible task. In addition to classroom use, the balanced coverage of theory and application also makes the book a valuable addition to the practitioner's bookshelf."

"The authors logically break down the discussion into four key coverage criteria: graph-based, logic-based, syntax-based, and partition-based testing. They provide a solid theoretical presentation of software testing and test coverage criteria … A concise but brief introduction to software testing."

"Software testing is one of the most important activities currently undertaken by our species, underpinning international security, social interaction, healthcare, transport, and economic well-being. No economic sector nor human activity remains untouched by software testing, yet it remains poorly understood, inefficient and often insufficiently effective, thereby requiring textbooks just such as this. This is an excellent introduction to software testing, covering principles, foundations, techniques and test management. It will be an invaluable book for practitioners, students and researchers alike."

"This book manages to capture - in a concise, clear, and precise form - the most important concepts pertaining to software testing. This is the best introduction to the topic that is available, as it includes many theoretical and practical insights that will provide the readers with a solid foundation in the subject matter."

"the best literature i have read on testing. Perfect combination of important topics and methods to get to the fifth level!"

Teaching evaluations and anonymous student comments

The following table presents Offutt's teaching and overall course evaluations at Mason, where 5 is the highest. At Mason, undergraduate course numbers reflect the year (205 is a sophomore course and 432 is a senior course). At the graduate level, 500-level and 600-level are predominately for MS students and 700-level courses are research-oriented, primarily for PhD students. In the CS department, scores above 4.5 in undergraduate courses and MS courses are rare, and above 4.75 is almost unheard of in senior-level courses.

Course	Semester	Teaching Evaluation	Course Evaluation
SWE 205	Spring 2018	4.88	4.76
SWE 301	Spring 2018	5.00	4.92
SWE 432	Fall 2015	4.85	4.71
SWE 437	Spring 2018	4.89	4.73
SWE 632	Fall 2016	4.62	4.60
SWE 637	Fall 2017	4.79	4.76
SWE 737	Spring 2017	5.00	4.83
SWE 763	Spring 2015	4.90	4.80

These numeric ratings reflect the universal regard for Offutt's courses. The excerpted comments below are more specific, detailed, and anecdotal.

- **SWE 205, Design Software Usability Analysis and Design**
  - This is a fun class!! I haven't figured out how this works. We were just talking and having fun, laughing and groaning and bad user interfaces, and didn't notice we were learning much. But then came the midterm and when I studied i thought "hey!!! when did we learn all this stuff?". every engineering should have to take this class.

- Really smart guy and technologically impressive. Interesting class discussions, relatively easy analysis oriented work throughout the class, useful knowledge gained. Nice guy.

- **SWE 432, Design and Implementation of Software for the Web**
  - A *great* teacher if you pay attention, keep up, and try. Kind of harsh if you get lazy. Very well organized and makes the lectures fun and interesting. You will NOT be sorry for taking his classes
  - This is the most fun class I've had since coming. Every CS major should be required to take this class!
  - I wish all my CS classes were like SWE 432, I enjoyed it a lot. Keep up the good work chief!!
  - The teacher is top notch, friendly teacher. I appreciate the acceration he gave the class in mid-year. Good call. PS A sense of humour is a admirable effortful and subtle teaching enhancement Thanks!
  - Professor Offutt has organized the materials of this course in a very effective manner. He is always willing to help and is very good in responding to students' questions. His jokes are not as funny as he wishes.
- **SWE 632, User Interface Design and Development**
  - You get out of this class whatever you are willing to put into it. Recommended for those interested in user-appropriate interfaces.
  - Dr. Offutt is hands down the best instructor I have had in SWE. His teaching is both entertaining and informative and his interest for the subject shows with his enthusiam in teaching it.
  - Answered all questions and went over material very thoroughly.
  - Dr. Offutt is on the few professors I've had who actually cares about teaching. That makes the biggest difference in the world! He was well organzied, the material was interesting. Overall, a good experience.
  - He is the best professor I have ever met!
  - Prof. Offutt always listens to the students and takes their contributinos to class discussions seriously. This allows for a comfortable give and take of ideas during class and a positive attitude toward the subject matter.
- **SWE 637, Software Testing**
  - Wow, what an awesome class, awesome book. It was not easy like I hoped, but taught me a lot about testing AND programming. Never saw anybody work so hard to get examples clear.
  - Great sincere professor
  - The best class I have taken at GMU. Prof Offutt returned all assignments the very next week. He is an excellent teacher - very organized, personable, made class fun and challenging.
  - Perhaps the best course I have had in my entire course sequence here at GMU. Excellent Professor. Best professor I have had in years.

Ethics in publishing

As editor-in-chief of Software Testing, Verification, and Reliability, Offutt writes an editorial every issue. In addition to introducing the papers, he often writes about general issues related to research and publishing[4]. One series of editorials (*Globalization—Ethics and Plagiarism*, *Plagiarism Is For Losers*, and *How the Web Resuscitated Evolutionary Design*) attracted a lot of attention for their simple and clear discussion of ethics in publishing. The online research magazine *Retraction Watch* interviewed Offutt and wrote a feature article about him and ethics, *Meet the scientist whose ideas were stolen at least three times*[5]. As a direct result of these writings, Aurali Dade, Associate Vice President Research Development, Integrity and Assurance at George Mason University regularly invites Offutt to talk with research students and new faculty at Mason's Responsible Conduct of Research Seminar.

PhD student symposiums at research conferences

One of Offutt's passions is helping young scientists. Thus, he is an eager volunteer organizer, panelist, and speaker at student symposiums at conferences. These allow PhD students to make short presentations of their ongoing research projects, and then get feedback from a panel of 3 or 4 professors. Some also feature presentations by some of the panelists. Offutt has participated in more than a dozen such symposia in the past decade, organized three, and spoken at several. His most popular talk is titled *How to get your paper rejected*, where he uses humor, humility, and real examples to describe all the reasons his papers have been rejected. His opening lines invariably create laughter and

[4] He archives his essays on his Mason website: https://cs.gmu.edu/~offutt/stvr

[5] On the web at https://retractionwatch.com/2018/01/15/meet-scientist-whose-ideas-stolen-least-three-times

draw the audience in: "*I have over 100 rejections. In fact, I'm confident I've had more papers rejected than anyone at this conference*." Powerpoint slides for this talk, and others, are posted on Offutt's university website.

Software for the community

As a PhD student in the 1980s, Offutt was a key architect and the primary programmer for a novel software testing research tool called Mothra. Offutt wrote more than half of the 200,000 lines of C programming language code (200K is very large for a university-built system, although medium-sized by industry standards). Mothra was distributed (on tape and then floppy disks!) with source code to hundreds of researchers and educators, who used it to support their classes and research projects. Offutt used it as his primary experimental lab for a decade after completing his PhD. Offutt has continued the tradition of making software tools that are robust enough to be useful to the community. A few are described below:

- **muJava** was built in the early 2000s as a research and educational mutation testing tool for the Java programming language. It is available as open-source and is still widely used, having been used by thousands of researchers and educators, and as a laboratory platform that has supported hundreds of research papers.
- **Godzilla** was an automatic test data generator that worked in conjunction with muJava to create tests cases. It was the first published tool to implement symbolic evaluation, and later dynamic symbolic evaluation, techniques that are now used in commercial tools such as Pex from Microsoft.
- **Graph coverage web application** is an online web app that supports ideas from Ammann and Offutt's book. It, along with its cousins **Logic coverage web app** and **Data flow coverage web**, is used by thousands of students, professors, and researchers to check their understanding of fundamental concepts from the testing book. These programs are hosted on servers run by Mason's Volgenau School of Engineering and accessible from the textbook website[6].
- **CoupTest** (Coupling-based Testing) is the result of an NSF-funded research project. It is a software analysis tool to extract coupling relationships and measure coverage of tests according to the coupling-based test technique.

Closing statement (first person)

In the mid-1990s, I was disenchanted with my research field of software testing. Too many decisions about papers and grant proposals were very personal and often unreasonably critical or unreasonably favorable. Conferences were dangerous minefields for young scientists where the wrong word to the wrong person could have career repercussions. Journal reviews were worse, with some senior scientists asking editors to review papers by specific individuals so they could punish them. I decided I had to either get out or fix it. Being stubborn, I chose to fix it.

It took many years to build the right network, achieve the stature needed, and understand how a rational academic field should operate. In 2008, I finally got the opportunity to become editor-in-chief of the premier journal in software testing (STVR). I took over with the explicit goal of "fixing" the review process to make it faster, fairer, and more positive. My next goal was to create a new major IEEE-sponsored conference. After two years of planning with Anneliese Andrews, Lionel Briand, and Benoit Baudry, all very successful software testing researchers, we created ICST. In its first year (2008), it was the biggest and most successful software testing conference ever. My most valued measure of success is that it draws young scientists into the field who all consider it a very positive experience.

Third, in 2008 Paul Ammann (of George Mason) and I published the first edition of a new textbook in software testing. *Introduction to Software Testing* filled a much-needed hole in the educational aspects of the field. With 80% market share, this book ensures that future software engineers and future researchers learn the right things about software testing in the right way.

Last year I was introduced at ICST's banquet as "*Jeff, the guy who ensured that all paths to software testing lead through George Mason University.*" Of course, I was proud to hear that. But what really matters to me is that young scientists today find the field of software testing to be a welcoming and nurturing environment.

[6] https://cs.gmu.edu/~offutt/softwaretest/