Dr. Gregory V. Wilson

Education and training:

Institution	Major	Degree	Year
Queen's University	Mathematics and Engineering	BSc	1984
University of Edinburgh	Artificial Intelligence	MSc	1986
University of Edinburgh	Computer Science	PhD	1993

Research and professional experience:

2012 – present	Project lead, Software Carpentry, Mozilla Foundation
2011	Software engineer, Side Effects Software Inc.
2010 – 2011	Independent training consultant, Software Carpentry
2006 – 2010	Assistant Professor, Dept. of Computer Science, University of Toronto
2004 – 2006	Independent contractor
2000 – 2004	Software engineer, Nevex/Baltimore Technologies/Hewlett-Packard
1998 – 2000	Independent contractor
1996 – 1998	Software engineer, Visible Decisions Inc.
1995 – 1996	Scientist, Centre for Advanced Studies, IBM Toronto
1992 – 1995	Post-doctoral researcher at various universities
1986 – 1992	Software engineer, Edinburgh Parallel Computing Centre
1985	Software engineer, Bell-Northern Research
1984 – 1985	Software engineer, Miller Communications Ltd.

Honors and awards:

2010	ComputerWorld Canada's IT Educator of the Year
2008	Jolt Award for Best General Book (with co-editor Andy Oram)
2004	University of Toronto Computer Science Student Union Teaching Award
1986	Howe Prize, University of Edinburgh
1985–86	Commonwealth Scholarship
1984	University Medal, Queen's University (top student in graduating class)
1984	A.B. Lillie Prize (top student in mathematics)
1982–84	Dean's Scholar, Faculty of Applied Science, Queen's University

Selected Publications:

* Amy Brown and Greg Wilson (eds.). *The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks*. Lulu, 2011.

* Andy Oram and Greg Wilson (eds.). *Making Software: What Really Works, and Why We Believe It*. O'Reilly, 2010.

* Jo Erskine Hannay, Hans Petter Langtangen, Carolyn MacLeod, Dietmar Pfahl, Janice Singer, and Greg Wilson. How do scientists develop and use scientific software? In *Proceedings of the Second International Workshop on Software Engineering for Computational Science and Engineering (SE-CSE 2009)*. IEEE, 2009.

* Greg Wilson. How do scientists really use computers? *American Scientist*, September/October 2009.

* Jordi Cabot and Greg Wilson. Tools for teams: A survey of web-based software project portals. *Doctor Dobb's Journal*, October 2009.

* Jennifer Campbell, Paul Gries, Jason Montojo, and Greg Wilson. *Practical Programming*. Pragmatic Bookshelf, 2009.

* David Matthews, Greg Wilson, and Steve Easterbrook. Configuration management for large-scale scientific computing at the UK Met Office. *Computing in Science and Engineering*, November/December 2008.

* Greg Wilson. Those who will not learn from history... *Computing in Science and Engineering*, 10(3), May 2008.

* Andy Oram and Greg Wilson (eds.). *Beautiful Code: Leading Programmers Explain How They Think*. O'Reilly, 2007.

* Jorge Aranda, Steve Easterbrook, and Greg Wilson. Requirements in the wild: How small companies do it. In *Proceedings of the 15th International Conference on Requirements Engineering*, October 2007.

* Greg Wilson. Where's the real bottleneck in scientific computing? *American Scientist*, January/February 2006.

* D. Winter, B. Vinegar, H. Nahal, R. Ammar, G. V. Wilson, and N. J. Provart. An 'electronic fluorescent pictograph' browser for exploring and analyzing large-scale biological data sets. *PLoS ONE*, 2(8), 2007.

* Greg Wilson. Open-source offers solutions for science software education. *Nature*, 436:600, July 2005.

* Greg Wilson. *Data Crunching: Solve Everyday Problems Using Java, Python, and More*. Pragmatic Bookshelf, 2005.

* Karen L. Reid and Gregory V. Wilson. Learning by doing: introducing version control as a way to manage student assignments. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, pages 272–276. ACM, 2005.

* Gregory V. Wilson and Paul Lu (eds.). *Parallel Programming Using C++*. MIT Press, 1996.

* Eshrat Arjomandi, William G. O'Farrell, and Gregory V. Wilson. Smart messages: An object-oriented communication mechanism for parallel systems. *Computing Systems*, 9(4):313–329, 1996.

* Gregory V. Wilson. What should computer scientists teach to physical scientists and engineers? *IEEE Computational Science and Engineering*, Summer and Fall 1996.

* Gregory V. Wilson. *Practical Parallel Programming*. MIT Press, 1995.

* Arthur Trew and Greg Wilson (eds.). *Past, Present, Parallel: A Survey of Available Parallel Computing Systems*. Springer-Verlag, 1991.

Synergistic activities:

Dr. Wilson co-founded Software Carpentry with Brent Gorda in 2004. Since then, he has grown it to an international volunteer program with over 100 instructors in a dozen different countries. In 2013 alone, Software Carpentry ran over 90 two-day workshops for almost 4500 scientists.

Dr. Wilson has consistently demonstrated a strong commitment to undergraduate education over more than two decades. He founded the Edinburgh Parallel Computing Centre's Summer Scholarship Programme, which recruited and trained 60 students between 1988 and 1992. While at the University of Toronto, he supervised over 150 undergraduates working alone or in small teams on almost 100 real-world projects, of which more than half were for clients outside the Computer Science department. In 2009, he created the UCOSP program, through which 90 students from more than a dozen Canadian universities worked in distributed teams on open source projects. He has been a mentor for Google's Summer of Code program since its inception in 2005.

Dr. Wilson has been a member of the Python Software Foundation since 2010.

Collaborators:
  • Marian Petre, Dept. of Computer Science, Open University (UK).

Graduated students and postdocs advised: Samira Abdi Ashtiani, Aran Donohue, Jeremy Handcock, Carolyn MacLeod, Jason Montojo, Rory Tulk.