

Dr. Gregory V. Wilson

Professional preparation:

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

Appointments:

2012 – present	Project lead, Software Carpentry, Mozilla Foundation
2011	Software engineer, Side Effects Software Inc.
2010 – 2011	Independent consultant
2006 – 2010	Assistant Professor, Dept. of Computer Science, University of Toronto
2004 – 2006	Independent consultant
2000 – 2004	Software engineer, Nevex/Baltimore Technologies/Hewlett-Packard
1998 – 2000	Independent consultant
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.

Products:

- * Software Carpentry: <http://software-carpentry.org>. Co-founded with Brent Gorda in 1998, this project's aim is to teach basic computing skills to researchers in science, engineering, medicine, and related fields. It now has over 100 volunteer instructors in a dozen countries, who delivered training to almost 4500 people in 2013 alone. All of Software Carpentry's lesson materials are freely available under the Creative Commons - Attribution license.
- * Amy Brown and Greg Wilson (eds.). *The Architecture of Open Source Applications*. Lulu, 2011. A series of books (the fourth volume is due to appear in 2014) in which the creators of major open source packages describe the architecture and performance of their software.
- * Andy Oram and Greg Wilson (eds.). *Making Software: What Really Works, and Why We Believe It*. O'Reilly, 2010. A collection of essays by leading researchers in empirical software engineering summarizing what we actually know about software development.
- * 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. The largest study ever done of how scientists use computers in their research.
- * Jennifer Campbell, Paul Gries, Jason Montojo, and Greg Wilson. *Practical Programming*. Pragmatic Bookshelf, 2009. A CS-1 introduction to programming using Python.
- * Andy Oram and Greg Wilson (eds.). *Beautiful Code: Leading Programmers Explain How They Think*. O'Reilly, 2007. A collection of essays from leading programmers about elegant software; winner of the 2008 Jolt Award for Best General Book.
- * Greg Wilson. *Data Crunching: Solve Everyday Problems Using Java, Python, and More*. Pragmatic Bookshelf, 2005.

Synergistic activities:

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 students from over a dozen Canadian universities work for a term in distributed teams on open source projects. For this and other work, Dr. Wilson was named ComputerWorld Canada's IT Educator of the Year in 2010.

Dr. Wilson has also been very active in the open source community for more than 15 years: He is a member of the Python Software Foundation, and has been a mentor for Google's Summer of Code program since its inception in 2005. He also works to build bridges between empirical software engineering research and industrial practice: he was an invited keynote speaker at SPLASH 2013, and a steward of "It Will Never Work in Theory" (<http://neverworkintheory.org>), an online forum for discussion of empirical results in software engineering that are of particular interest to practitioners. As part of this work, he has initiated and edited a series of book connecting theory to practice, including *Beautiful Code*, *Making Software*, and a multi-volume series titled *The Architecture of Open Source Applications* (available at <http://aosabook.org>).

Collaborators:

Marian Petre (Open U.); Amy Brown (independent contractor); Eleni Stroulia (U. Alberta); Ken Bauer (Tecnologico de Monterey); Michelle Craig (U. Toronto); Karen Reid (U. Toronto); Andy Oram (O'Reilly Media); D.A. Aruliah (University of Ontario Institute of Technology); C. Titus Brown (Michigan State U.); Neil P. Chue Hong (Software Sustainability Institute); Matt Davis (Datapad, Inc.); Richard T. Guy (Microsoft); Steven H.D. Haddock (Monterey Bay Aquarium Research Institute); Kathryn D. Huff (U. California - Berkeley); Ian M. Mitchell (U. British Columbia); Mark D. Plumbley (Queen Mary University London); Ben Waugh (University College London); Ethan P. White (Utah State U.); Paul Wilson (U. Wisconsin - Madison).

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