Alex Groce (agroce@gmail.com), Northern Arizona University

Robert K. Merton's *On the Shoulders of Giants: A Shandean Postscript - The Post-Italianate Edition*, or, henceforth, OTSOG, is our (ostensible) topic for this issue's *Passages* column.

[[What follows after this double-bracketed "introduction" is a failure, an attempt to recommend this book by, in miniature, aping its style and form. In my opinion, it doesn't really work. I ask the reader who has ever enjoyed a book read because it was recommended in this column to bear with me through it, however, for there is a point to be made, and a remarkable book to recommend.

Laurence Sterne wrote *Tristram Shandy* under a sentence of death, the slow but inexorable progression of tuberculosis; his oddity of a novel has survived him by centuries, though after its initial great popularity, it was largely forgotten, which prompted Dr. Johnson to remark that "Nothing odd will do long. Tristram Shandy did not last." For once, Johnson was wrong. Sterne included himself in the book, as Yorick, who dies early in the "narrative" (such as it is), though he also appears on its last page.]]

The famous aphorism of Sir Isaac Newton that is the ostensible topic of Merton's book (and which gives it its' title), taken from a letter to Hooke (a man far less-well known than Newton, but recently perhaps more celebrated due to his prominence in Neal Stephenson's Baroque Cycle of novels [which are concerned, fundamentally, with Newton himself, of course, as any books documenting and, to be sure, considerably punching-up, the birth of modern science, would be]) wherein he wrote, "If I have seen further it is by standing on the sholders [*sic*] of Giants" (Newton did not write "[sic]", of course), is probably well-known to most educated readers.

That the Aphorism is relevant to computer science, and software engineering, in particular, also needs little defense. In 2017, according to https://www.acm.org/binaries/content/assets/publications/advertising/acm-audience-profile.pdf, 23% of the ACM media audience were academic researchers, and another 30% were college professors. Or, rather, 23% were academic researchers, and 30% were college professors. At first glance, the relevant table appears to show that 53% of ACM media readers (and we are just going to assume SIGSOFT SEN demographics are similar to overall ACM audience, though of course this is simply a guess, without the kind of rigorous support that sure scholarship would have; however, this is a book review column, not serious scholarship or peer-reviewed research) are academics of some sort, but it turns out the numbers in the table add up to 174%, due to people selecting multiple job responsibilities. Still, at least 30% of ACM readers, including, we are going to guess, readers of this column, are professors, and all professors are quite familiar with Google Scholar, both because it is very useful and because, to our shame, many of us like to look up our own statistics (mine are here: https://scholar.google.com/citations?user=ewrrvq8AAAAJ&hl=en) and see how we are faring, both compared to our former selves (how many new cites *did* I get this year?) and compared to our colleagues. The motto of Google Scholar? "**Stand on the shoulders of giants.**"

OTSOG!

I say “to our shame,” but, of course, narcissism of a sort can, like many ignoble things, also serve some greater good. The mere pursuit of citations as an end-in-itself can be a wasteful thing, an artificial and pointless numbers-game, but it is also true that, to a large extent, the purpose of science, and particularly public science as performed by that 30% of us who are professors, is to *communicate new knowledge*. One way to measure the extent to which one has succeeded in this is to see how much other new knowledge is built, as it were, by standing on one’s shoulders (that is, to measure the current weight of dwarves using you as a stepladder, a rather painful metaphor if taken too literally). One obvious way to maximize this number is to do good work that will be of interest to other people, so the means of selfishness and self-aggrandizement can, sometimes, be turned to ends that are largely, probably, beneficial to the public. This is certainly a fitting idea to bring up in discussing a book (OTSOG) that is centered on a time period whose diameter, though Newton died when Adam Smith was 4 years old, includes the period during which Smith flourished, and refers to that sage of how a system may transform private ends of a non-virtuous sort (in a note on page 95 [there are a lot of notes in this book, and they are helpfully on the pages, not relegated to an inconvenient end-note form as in David Foster Wallace’s *Infinite Jest*]), so long as they are embedded in a context of virtuous constraints, into public ends of a most beneficial sort. For science, of course, as business and commerce, the public context of education in virtue and inculcation of limits to means used to achieve selfish ends, is critical; scientists who seek to maximize citation count and lack integrity will be tempted to fake data and distort their results to obtain popularity. That this is all too common (along with simple human fallibility and gullibility, plus the desire for a cracking good surprising result, as well as more technical issues such as are often ably discussed by Andrew Gelman: https://statmodeling.stat.columbia.edu/2018/02/18/low-power-replication-crisis-learned-since-2004-1984-1964/) has resulted in the ongoing replication crisis, which is not so disastrous in computer science and software engineering as in, say, the social sciences or medicine, but is still troubling (at least to this author).

This notion, of faking it to make it, due to lack of ethical concerns and moral education (or simply being a very nasty person, since as Fielding remarks in *Joseph Andrews*, though I cannot place the exact quotation or idea, there may be some horses that are simply bad [my inability, even with the help of search in the Project Gutenberg text, to identify the precise passage, is unfortunate, though Ezra Pound remarked that “the domain of culture begins when one HAS ‘forgotten-what-book’” and forgetting what exactly in what book is one step towards that point]) is of course also important in software engineering. We generally, these days, require that graduating students have some class on ethics in computer science, and the ACM Code of Ethics of course has no tolerance for lack of integrity. “Honesty is an essential component of trustworthiness. A computing professional should be transparent and provide full disclosure of all pertinent system capabilities, limitations, and potential problems to the appropriate parties. Making deliberately false or misleading claims, fabricating or falsifying data, offering or accepting bribes, and other dishonest conduct are violations of the Code,” as it says on page 7 the Code (https://www.acm.org/binaries/content/assets/about/acm-code-of-ethics-booklet.pdf). My own particular sub-field, software testing, relies on developers and testers to try their best to break their software and forth-rightly present the findings from that effort, not hide defects or fail to look for them effectively in an attempt to curry favor or meet deadlines. Corner cutting, even without conscious evil intent, can be fatal, as the Therac-25 tragedy (https://hackaday.com/2015/10/26/killed-by-a-machine-the-therac-25/) shows:

> "AECL never publicly released the source code, but several experts including [Nancy Leveson] did obtain access for the investigation. What they found was shocking. The software appeared to have been written by a programmer with little experience coding for real-time systems. There were few comments, and no proof that any timing analysis had been performed. According to AECL, a single programmer had written the software based upon the Therac-6 and 20 code. However, this programmer no longer worked for the company, and could not be found."

More generally, the Aphorism is applicable to the work of software engineering, not just to self-absorbed, if beneficial, academics/scholars. Learning practices, like those developed in safety engineering since the Therac-25 incident, is a process whereby each software engineer's capability is greatly multiplied by the efforts and discoveries of previous software engineers. One great advantage of software engineering is that the giants-as-footstools metaphor can be reified by the creation of tools, software systems that function as giants for us dwarves. If we have been more productive in software development tasks, it is because we have stood on the shoulders of millions of lines of code that support our work, while the Ancients, such as Grace Hopper, had to come up with the very idea of a compiler and other foundations we simply take for granted. There is (the idea of the Passages columns was in part originated in my personal desire to fight it) perhaps a tendency in computer science to give lip service to the Giant-ness of figures such as Hopper, or Lovelace, or Babbage, or Turing, or still-living giants such as Ken Thompson or Knuth or David Parnas or Fred Brooks or Mary Shaw, but mostly to act as if any idea older than the latest reactive web framework or agile methodology (are we still into agile?), or fuzzing technique, is boring and trite. We know so much more than they; but they are what we know (to paraphrase T. S. Eliot in "Tradition and the Individual Talent" remarking that

> "Some one said: `The dead writers are remote from us because we know so much more than they did.` Precisely, and they are that which we know."

But our space is nearly full and we have not yet really talked about OTSOG. Robert K. Merton's OTSOG is a book that resembles this column, as a canon by Bach resembles "Row, Row, Row Your Boat," except that the latter is at least something that has endured for a very long time, and certainly has a charm. OTSOG is a digressive, roundabout, much-footnoted parody of a scholarly book, delighting in its imitation and flattery of Laurence Sterne's metafictional and baroque novel *The Life and Opinions of Tristram Shandy, Gentleman*. However, under the endlessly playful surface, Merton has written a book that is profoundly insightful about issues of scientific priority, the sociology of science, the transmission of knowledge, and the nature of the debt we owe to our historical predecessors in any intellectual enterprise that spans generations. Software engineering, as a body of knowledge and tradition (a tradition is a “river of light”, is inherently concerned with these ideas, and I cannot imagine anyone who made it this far in this column not being entranced and enlightened by spending time with Merton’s great and famous book. Also, Merton is pretty darn funny, when he isn’t giving profound insights into the nature of science, knowledge, authority, innovation, and, most of all (see below) gratitude (and even more so when he is). A “citation” can be a “much thanks.”

Notes:

1. The author of this review does not mention where he came upon the existence of OTSOG, which seems a non-Mertonish way to go about things. The answer is directly via a letter in *Questioning Minds*, a giant compendium of letters between Hugh Kenner and Guy Davenport (https://lareviewofbooks.org/article/a-mutual-bewitchment/ is a good review, https://newrepublic.com/article/153244/hugh-kenner-guy-davenport-letters-unlikely-literary-friendship an even better one), on pages 1025 and 1026, which probably give a better idea of the greatness of OTSOG than this review; indirectly, he probably attended to that intriguing mention because he loves *Tristram Shandy* so much, which the author owes, if memory serves, to the generous and perceptive writer and teacher John Kessel strongly recommending it to him in the course of a class on fantasy literature at North Carolina State University, many years ago.

[[The point to be made, for which you have been waiting, is this. Late in this book, Merton relates the story of how a gathering of great physicists prompted one to note, first, that most scientists who have ever lived are now living, and second that we have the unique opportunity to speak with the giants on whose shoulders we stand. For all its formal perversity and oddity, OTSOG is permeated with a vision of the complexity of the transmission of knowledge, and the debts we owe to others for our achievements and insights. Like Sterne, we all write our code under sentence of death, and it would be well to talk now, while we have a chance, to the giants who yet live among us, and tell them of our gratitude. The comedy of transmission, of accidental use of an answer from stackoverflow in a context where it does not work, of inspiration by the error of another, is a comedy also because it is a series of marriages of minds, through ideas and through code and through systems, across many lives, both horizontally in the now and vertically in time.

That is, if Brian Kernighan writes a memoir, you should read it, even if you think you know plenty about modern UNIXen. If Fred Brooks, Barbara Liskov, or Andreas Zeller gives a talk at a nearby university, go. It’s too late to hear John Reynolds. If you think of writing a “fan email” to some luminary whose light has been a road for you, write that email now (if it’s for Knuth, you’ll need to write an actual letter).]]