



ISTOCKPHOTO

One of the things I adore about the field of software testing is how intertwined it is with that of philosophy. Substantial discussion of testing very quickly segues into such philosophical domains as aesthetics, epistemology, and ethics.

This may at first seem like a surprising collision. Philosophy? Isn't that ivory-tower stuff, where bearded academics argue endlessly over trees falling in the forest? Testers, on the other hand, live in the metaphorical trenches of the software industry, pressing our faces up against the code, scrutinizing it for flaws, and racking our brains for ways to coax the software we're testing into giving up its secrets. What could such a technical field possibly have in common with one that's firmly seated amongst the humanities?

Inquiry is the core of software testing—inquiry into how the software behaves, inquiry into how people believe it should behave, and, hopefully, inquiry into how best to inquire. Philosophy is the archetypical process of inquiry. It provides both the framework for how we think of inquiry and

the means we use to carry it out. If we are to refine the way we go about our work, then the tools of philosophical analysis are indispensable.

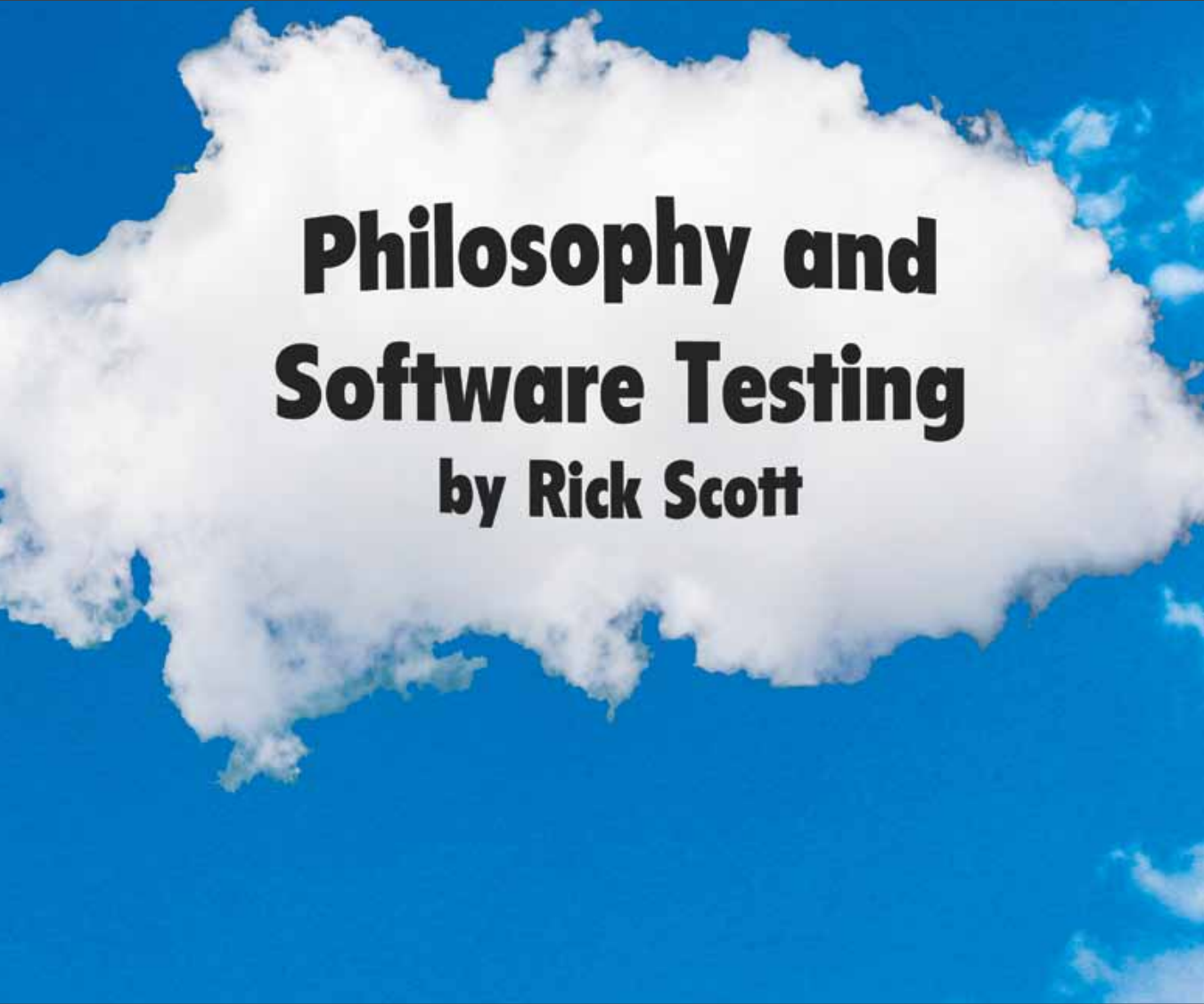
Branches of Philosophy

There is no particularly neat way to divide the field of philosophical inquiry; a great deal of overlap occurs regardless of how the lines are drawn. That being said, one classical set of divisions runs through the following branches: ethics, metaphysics, epistemology, logic, and aesthetics.

ETHICS

Ethics addresses questions of morality. What actions are right and wrong? By whom can one do right or wrong? By what standard should we evaluate the morality of choices or behaviors? Given a choice between two actions both damnable, what should we do?

Ethical questions often come into play in software testing, and unfortunately they're not always given the consideration they deserve.



Philosophy and Software Testing

by Rick Scott

As testers, what should our ethical code be? What are our responsibilities to the owner of the software under test? To the software's users? To our teammates? While it's easy to pontificate on what the right thing to do is, living up to that standard is another thing entirely.

METAPHYSICS

Metaphysics is the study of reality and what is real. On the face of it, this seems laughable. Don't we all take reality for granted? How would we go about our day if we didn't believe in the existence of the world we perceive? However, metaphysics also delves into things that are not so readily observable, like causality, ontology (the study of categories), and possibility.

If the software exhibits a bug during testing, how do we know our actions precipitated it and not something else? When is it appropriate to classify features, users, or elements of a design as similar? If a risky and untested change is put into production but nothing goes wrong, is there still any basis for arguing that it was a bad idea?

EPISTEMOLOGY

Epistemology is the study of knowledge and knowing. What is knowledge? How can we tell what we know and what we don't? Is observation a valid way of acquiring knowledge? What about reason or feelings?

Since testing is, at its core, a practice of acquiring knowledge about how a piece of software works, some of these questions may sound very familiar. When do we stop testing, knowing that we know enough about the state of this piece of software to do so? How can we separate what we know about the software from what we merely believe?

LOGIC

We think of logic as being the branch of philosophy most closely related to the field of software. Formal logic—mathematical logic in particular—dictates how hardware gates, software control structures, and database queries work. It gives us tools like truth tables and Boolean algebra, which we can then use to analyze possibilities, decisions, or the flow of a program.

However, just as the importance of humans in software is frequently overlooked, so, too, is the importance of informal or persuasive logic. Software creators, software users, and software project managers are people, and the importance of being able to interact with all of them in a reasonable way cannot be overstated. Can you build a case for making a certain technical decision? How about formulating a reasonable approach when it comes to encountering a bug in the wild or justifying that approach later? Informal logic is as crucial to navigating the human and organizational side of making software as formal logic is to interfacing with computers.

AESTHETICS

“The study of beauty” is the most commonly offered description of aesthetics, but this definition is as superficial as it is glib. Aesthetics addresses how and why we can make meaningful critical judgments on whether or not something possesses beauty, elegance, or value—judgments that are always to some degree subjective.

This all sounds well and good, but what does it have to do with the world of computing? We tend to think of our work as being technical, mechanical, and mathematical; as having cut and dried answers; as being far removed from the subjective worlds of art and music. But, I would argue that our work should exemplify beauty as well, and not just in the areas of design or UI, where it is most obvious. Well-made software exhibits what open source hackers refer to as elegance—a combination of simplicity, grace, and power.

Programs that are well organized and straightforward to understand are better than impenetrable spaghetti code. Interfaces that are clean and easy to use are better than those that are cluttered and require contortions on the part of the user. Study of aesthetics helps one discover and articulate consistent principles behind why code written a certain way is clearer or why organizing an interface a certain way makes things easier for users.

Whether we explicitly acknowledge it or not, philosophy informs the key decisions surrounding software and its creation. Its study hones the skills of reasoning, inquiry, and analysis that are at the core of software testing. Familiarity with philosophy allows one to place software in a larger context and ask critical questions about it—questions as broad as what role one’s code plays in society, and as focused as how users will get enough value out of it to make it worth paying money for. Thus, a fundamental understanding of philosophy is an indispensable aid for any software professional. **{end}**

rick@shadowspar.dyndns.org

This article is the first installment in a series exploring the overlap of ideas between software testing and the philosophic fields of ethics, metaphysics, epistemology, logic, and aesthetics. Visit TechWell.com to read Rick Scott's other articles in the series.

Better Software Magazine Publisher's Statement

October 2011

Published in Issue 13-6 November/December 2011
UNITED STATES POSTAL SERVICE

Statement of Ownership, Management, and Circulation

1. Publication title: Better Software
2. Publication number: 0019-578
3. Filing date: October 2011
4. Issue frequency: January/February, March/April, May/June, July/August, September/October, November/December
5. Number of issues published annually: 6
6. Annual subscription price: \$19.95
7. Complete mailing address of known office of publication: Software Quality Engineering, 340 Corporate Way, Ste. 300, Orange Park, FL 32073-6214, Clay county
8. Complete mailing address of headquarters or general business office of publisher: Software Quality Engineering, 340 Corporate Way, Ste. 300, Orange Park, FL 32073-6214, Clay county
9. Full names and complete address of Publisher, Editor, and Editor in Chief: Publisher: Wayne Middleton, Software Quality Engineering, 340 Corporate Way, Ste. 300, Orange Park, FL 32073-6214, Clay county. Editor in Chief: Heather Shanholtzer, Software Quality Engineering, 340 Corporate Way, Ste. 300, Orange Park, FL 32073-6214, Clay county.
10. Owner: C. Wayne Middleton, Software Quality Engineering, 340 Corporate Way Ste. 300, Orange Park, FL 32073-6214, Clay county.
11. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages, or other securities: None.
12. NOT NON PROFIT; DO NOT NEED TO INCLUDE IN STATEMENT
13. Publication title: Better Software
14. Issue date for circulation data below: September/October 2011
15. Extent and Nature of Circulation:
 - a. Total number of copies (Net press run). Average Number Copies Each Issue During Preceding 12 Months: 4984. Actual number of copies of single issue published nearest the filing date: 6985.
 - b. Legitimate Paid and/or Requested Distribution (By Mail and Outside Mail) (1b) Outside County Paid/Requested Mail Subscriptions Stated on PS Form 3541: Average number of copies each issue during preceding 12 months: 3863. Actual number of copies of single issue published nearest the filing date: 5103. (2b.) In-County Paid/Requested Mail Subscriptions stated on PS Form 3541: Average number of copies each issue during preceding 12 months: 0. Actual number of copies of single issue published nearest the filing date: 0 (3b) Sales through dealers and carriers, street vendors, counter sales, and other non-USPS paid distribution as stated on PS Form 3541: Average number of copies each issue during preceding 12 months: 184. Actual number of copies of single issue published nearest the filing date: 295 (4b) Requested Copies Distributed by other mail classes Through the USPS (e.g. First-Class Mail). Average number of copies each issue during preceding 12 months: 0. Actual number of copies of single issue published nearest the filing date: 0
 - c. Total paid and/or requested circulation [Sum of 15b. (1), (2), (3), and (4)]. Average number of copies each issue during preceding 12 months: 4047. Actual number of copies of single issue published nearest the filing date: 5398.
 - d. Nonrequested Distribution (By Mail and Outside Mail). (1d.) Outside-county Nonrequested Copies Stated on PS Form 3541. Average number of copies each issue during preceding 12 months: 144. Actual number of copies of single issue published nearest the filing date: 1. (2d.) In-county Nonrequested as stated on form 3541. Average number of copies during preceding 12 months: 0. Actual number of copies of single issue published nearest the filing date: 0. (3d.) Nonrequested copies distributed through the USPS by other classes of mail (e.g. First class, non-requestor Other classes mailed through the USPS. Average number of copies each issue during preceding 12 months: 0. Actual number of copies of single issue published nearest the filing date: 0. (4d.) Non requested Copies distributed outside the mail (include pickup stands, trade shows, showrooms, and other sources) Outside the Mail (other carriers). Average number of copies each issue during preceding 12 months: 761. Actual number of copies of single issue published nearest the filing date: 1,395.
 - e. Total Nonrequested Distribution. Average number of copies each issue during preceding 12 months: 905. Actual number of copies of single issue published nearest the filing date: 1396.
 - f. Total distribution (Sum of 15c. and 15e.). Average number of copies each issue during preceding 12 months: 4952. Actual number of copies of single issue published nearest the filing date: 6794.
 - g. Copies not distributed. Average number of copies each issue during preceding 12 months: 32. Actual number of copies of single issue published nearest the filing date: 194.
 - h. Total (Sum of 15f. and g.). Average number of copies each issue during preceding 12 months: 4984. Actual number of copies of single issue published nearest the filing date: 6985.
 - i. Percent paid and/or requested circulation (15c. divided by 15f. times 100). Average number of copies each issue during preceding 12 months: 81.7%. Actual number of copies of single issue published nearest the filing date: 79.5%.
16. Publication of Statement of Ownership is printed in the November/December 2011 (13-6) issue.

I certify that the statements made by me above are correct and complete.

Heather Shanholtzer, Editor in Chief