Test Suites: A Tool for Improving Student Articles

Eugene Volokh

The Problem

Students who are writing law review notes or seminar papers often get tunnel vision: they focus on the one situation that prompted them to write the piece—usually a situation about which they feel deeply—and ignore other scenarios to which their proposal might apply. And this often leads them to make proposals that, on closer examination, prove to be unsound.

For instance, a student might be outraged by the government's refusal to fund abortions, and might therefore propose a new rule that "if the government funds the nonexercise of a constitutional right, then the government must also fund the exercise of the right"; or the student might simply propose that "if the government funds childbirth, it must fund abortions," and give the more general claim as a justification. But the student might not think about the consequences of this general claim, which are that when the government funds public school education, it must also fund private school education (since that's also a constitutional right)—or perhaps even that a government that funds antidrug speech must also fund prodrug speech.

The student's argument, at least at its initial level of generality, is thus likely wrong or at least incomplete even by the student's own lights. But the focus on the one core case keeps the student from seeing the error.

All of us have run into this in our students, and we've tried to help them by identifying the counterexamples that they need to consider—and by stressing to them that they should themselves identify such counterexamples. I want to suggest a more systematic approach for doing this, using a concept borrowed from computer programming: the *test suite*.

A test suite is a set of cases that programmers enter into their programs to see whether the results look right. A test suite for a calculator program might contain the following test cases, among many others:

1. Check that 2+2 yields 4.
2. Check that 3-1 yields 2.
3. Check that 1-3 yields -2 (because the program might work differently with positive numbers than with negative ones).

Eugene Volokh <volokh@law.ucla.edu> is a professor of law at the University of California, Los Angeles, and the author of *Academic Writing: Student Notes and Seminar Papers* (forthcoming 2003, Foundation Press), from which this article is drawn.

Journal of Legal Education, Volume 52, Number 3 (September 2002)

4. Check that 1/0 yields an error message.

If all the test cases yield the correct result, then the programmer can have some confidence that the program works. If one test yields the wrong result, then the programmer sees the need to fix the program—not throw it out, but improve it. Such test suites are a fundamental part of sound software design. Before going into law, for instance, I wrote a computer program that had 140,000 lines of code and 50,000 lines of test suites.

Students can use a similar approach for testing legal proposals. I tell my students that *before* they commit themselves to a particular proposal, they should design a test suite containing various cases to which their proposal might apply.[1]

Assume, for instance, that a student is upset by the way peyote bans interfere with some American Indian religions. The government has no business, the student wants to argue, imposing such paternalistic laws on religious observers. This student should design a set of test cases that involve requests for religious exemptions from many different kinds of paternalistic laws, for instance:

1. requests for religious exemptions from assisted-suicide bans, sought by a doctor who wants to help a dying patient die, or by the patient who wants a doctor's help
2. requests for religious exemptions from assisted-suicide bans, sought by someone who wants to help physically healthy fellow cult members commit suicide
3. requests for religious exemptions from bans on the drinking of strychnine (an example of an extremely dangerous activity)
4. requests for religious exemptions from bans on the handling of poisonous snakes (an example of a less dangerous activity)
5. requests for religious exemptions from bans on riding motorcycles without a helmet (an example of a less dangerous activity, but one that—unlike in examples 3 and 4—many nonreligious people want to engage in)[2]

Then, once the student designs a proposed rule, he should test it by applying it to all these cases and seeing what results the proposal reaches.

1. See, e.g., Jennifer Rothman, Freedom of Speech and True Threats, 25 Harv. J.L. & Pub. Pol'y 283, 336 (2001), for an example of one such test suite that the student used while writing her article and eventually incorporated into the published version.

2. This discussion builds on Eugene Volokh, Intermediate Questions of Religious Exemptions—A Research Agenda with Test Suites, 21 Cardozo L. Rev. 595 (1999); for examples of the incidents on which the test suite is based, see *id.* at 603 n.18 & 630 nn.106–109. *Cf.* KDM *ex rel.* WJM v. Reedsport School Dist., 196 F.3d 1046, 1056–57 (9th Cir. 1999) (Kleinfeld, J., dissenting) (also using computer test suites as a model for testing legal claims).

What Students Might Find by Testing Their Proposals

What information can this testing provide?

1. Error. The student might find that the proposal reaches results that even he thinks are wrong. For instance, suppose the proposal is that religious objectors should always get exemptions from paternalistic laws. Thinking about test case 2 might lead the student to conclude that religious objectors should not be allowed to help physically healthy people commit suicide. The proposed rule, then, would be unsound even by the student's own lights.

What can the student do about this?

The student might think that the proposal yielded the wrong result because it didn't take into account countervailing concerns that may be present in some cases—for instance, the special need to prevent even a voluntarily assumed near-certainty of death or extremely grave injury, rather than just a remote risk of harm. If this is so, he could *modify* the proposed test, for instance by *limiting its scope* (e.g., by adding an exception for harms that are likely to be immediate, grave, and irreversible).

Another possibility is that the insight which led the student to suggest the proposal—in our example, the belief that there should be a religious exemption from peyote laws—is better explained by a *different rule.* For instance, as the student thinks through the test cases, he might conclude that the real problem with the peyote ban is that it's factually unjustified (because peyote isn't that harmful), and not that it's paternalistic. The student might then *substitute* a new rule: courts should allow religious exemptions from a law when they find that the religious practice doesn't cause any harm, whether or not the law is paternalistic.

2. Vagueness. The student might find that the proposal is unacceptably vague. Say that the proposal was that religious objectors should be exempted from paternalistic laws when "the objectors' interest in practicing their religion outweighs the government's interest in protecting people against themselves."

In the peyote case, this proposal might have satisfied the student, because it was clear to him that the government's interest in protecting people against peyote abuse was so weak. But as he applies the proposal to the other cases, he might find that the proposal provides far too little guidance to courts—and might therefore lead to results he dislikes. This could be a signal for the student to *clarify* the proposal.

3. Surprise. The student might find that the proposal reaches a result that he at first thinks is wrong, but then realizes is right. For instance, before applying the proposal to the test suite, the student might have assumed that religious objectors shouldn't get exemptions from assisted-suicide bans. But after he thinks more about this test case in light of his proposal, he might conclude that his intuition about assisted suicide was mistaken.

The student should keep this finding in mind and *discuss it in the article:* it may help him show the value of his claim, because it shows that the proposal yields counterintuitive but sound results.

4. Confirmation. The student might find that the proposal precisely fits the results that he thinks are proper. This should make the student more confident in the proposal's soundness; and it also provides some examples that he can *use in the article to illustrate* the proposal's soundness (as I discuss below).

Developing the Test Suite

How can students identify (perhaps with your help) good items for their test suites? Here are a few suggestions that I give students

1. Start by identifying what needs to be tested. The test suite is supposed to test the proposed legal principle on which the claim is based. Sometimes the claim is itself the principle. For instance, if the proposal is that "the proper rule for evaluating requests for religious exemptions from paternalistic laws is [thus-and-such]," the student would need a set of several cases to which this rule can be applied.

But sometimes the claim is just an application of the principle; for instance, the claim that "religious exemption requests from peyote laws should be granted" probably rests on a broader implicit principle that describes which exemption requests should be granted. If that's so, then the student should come up with a set of cases that test this underlying principle. One case should involve peyote bans, but the others shouldn't.

2. Each test case should be plausible. It should be the sort of situation that might happen in real life. It's good to base it on a real incident, whether one drawn from a reported court decision or a newspaper article. The student need not precisely follow the real incident, and may assume slightly different facts if necessary; the goal is to have the reader acknowledge that the case *could* happen the way it's described, not that it necessarily has happened. But the student should make sure that any alterations still leave the test case as realistic as possible.

3. The test suite should include the famous cases in this field. This can help confirm for the student and the readers that the proposal is consistent with those cases—or can help explain which famous cases would have to be reversed under the proposal.

4. At least some of the cases should be challenging for the proposal. The student should identify cases where the proposal might lead to possibly unappealing results, and include them in the test suite. I point out to my students that skeptical readers, including me, will think of these cases eventually. Identifying the hard cases early—and, if necessary, revising the proposal in light of them—is better than having to confront them later, when changing the paper will require much more work.

5. The test cases should differ from each other in relevant ways, since their role is to provide as broad a test for the claim as possible. If the student is testing a claim about paternalistic laws, for instance, he shouldn't just focus on various drug laws, or just on paternalistic laws aimed at protecting children. The student should think of many different sorts of paternalistic laws, and choose one or two of each variety.

6. The cases should yield different results. For instance, if the student's proposed rule judges the constitutionality of a certain type of law, he should find some laws that he thinks should be found unconstitutional, some that he thinks should be found constitutional, and some whose constitutionality is a close call.

7. The cases should involve incidents or laws that appeal to as many different political perspectives as possible. Say that the student is a liberal who wants to argue that the Free Speech Clause prohibits the government from funding viewpoint-based advocacy programs. He might have developed this view because he thinks the government shouldn't be allowed to fund antiabortion advocacy, and his proposal will indeed reach the result he prefers in that case.

But what about advocacy programs that liberals might especially favor, such as prorecycling advocacy, or advertising campaigns promoting tolerance of homosexuals? It would help if the test suite included such cases, plus generally popular programs such as antidrug advertising, or programs that even small-government libertarians might like, such as advocacy of respect for property rights (for instance, antigraffiti advocacy). This wide variety of test cases will help show the student whether the proposal is indeed sound across the board, or whether even he himself would on reflection oppose it.

Testing as an Iterative Process

When one of my students designs a test suite, I check to make sure that it's thorough enough, and then I have the student apply the proposal to the test suite. The student may find some flaws in the proposal and update the proposal in light of his discoveries. Then I tell him to apply it to all the test cases again: maybe the revision fixed the proposal for one test case, but broke it for another. More broadly, whenever the student updates the proposal as he is writing his article, he should make sure that it still yields the right results in all the test cases.

Using the Test Suite as Part of the Article

The test suite is the student's tool for proving to himself and to you that his claim is sound. It can also be a tool to prove the same to readers. After presenting the proposal, the article should show the reader how the proposal applies to a variety of examples drawn from the test suite.[3] There are three advantages to this.

1. This application will help make the proposal clearer and more concrete for readers.
2. It can help prove to the reader that the proposal reaches the right results.
3. Applying the proposal to the test suite in writing can help the student make sure that the proposal does indeed reach the right results.

3. See, e.g., Eugene Volokh, Freedom of Speech, Shielding Children, and Transcending Balancing, 1997 Sup. Ct. Rev. 141, 183–87.

Of course the student doesn't have to use every test case in the suite; some might prove redundant or only marginally relevant. But the test suite can provide a starting set of concrete examples that can help the student make the proposal clearer and more persuasive to readers.

The Role of the Law Review

I have suggested that creating the test suite should be a mandatory, and early, step in a student's writing project. You can enforce this requirement as a teacher, but the law review can also enforce it through its own notes department.

Third-year law review editors often work closely with second-year staffers to guide them through the note-writing process, and I think they should insist on test suites even if the student's faculty adviser doesn't much care about them. This insistence would be a *service to the staffers,* for the reasons I describe above. But beyond this, the law review should feel a *professional obligation* to make sure that the student notes that it publishes are as sound as possible—and thorough test suites can help with this process.

Summary: Who Wins and Why

Test suites, then, are a valuable tool, for students, faculty advisers, law review editors, and readers.

1. They help students think through the problem concretely, with an eye toward actual applications of their proposal.

2. They help students identify unexpected consequences (both appealing and unappealing) of their proposal.

3. They help students avoid focusing just on their own pet cases and ignoring the other cases to which the proposal might be applied.

4. They let students find and solve potential problems at the beginning of the process, rather than after they finish drafts and submit them to their advisers or other readers.

5. They save faculty advisers time by helping students find more potential problems and counterarguments on their own.

6. They help make law review notes more logically sound, more concrete, and thus more helpful to lawyers, academics, and judges.

They can do all this with only a modest amount of extra upfront work on the student's part; and this extra work at the beginning might actually reduce the amount of work that the student will need to do later.