I Speak For The User: The Problem of Agency in Software Development

Cem Kaner, J.D., Ph.D.

Many testers see themselves as user advocates and get into passionate arguments over bugs they consider unacceptable to the user. The bugs might be in the code, the visible design, the interoperability with other devices or systems, etc. Other members of the team might disagree with the assessment of the bug ("That's not a bug, it's a feature!") or its priority.

At some point, some testers get incensed or insistent. They feel that the development team has to listen to them because Testers Speak For The User.

There are a couple of interesting questions here.

First, testers aren't the only people who have the "We Speak For The User" message drummed into them at almost every conference or workshop. Writers hear this a lot too. So do tech support / help desk / field support staff. Usability testers. Installers. System administrators. Marketers. (Yes, really. Marketers.) Salespeople. (Who do the users tell "no, I don't like it and I won't pay for it?" Salespeople! Who has the strongest vested interest in getting it right for the user? Hmmm…)

So when some tester protests in a meeting,

> "But you have to fix this. The user will hate it. I know the user will hate it because… I Speak For The User"

what message does that send to all those other people who think "We Speak For The User"?

Who's right?

That brings us to the second question. When Person $\mathcal{A}$ say that s/he speaks on behalf of person $\mathcal{U}$, that $\mathcal{A}$ can decide whether something is acceptable for $\mathcal{U}$ or not, and $\mathcal{A}$'s decisions are binding on $\mathcal{U}$, then $\mathcal{A}$ is speaking as $\mathcal{U}$'s *agent*.

How does $\mathcal{A}$ become $\mathcal{U}$'s agent?

$\mathcal{U}$ (or an agent of $\mathcal{U}$) has to appoint $\mathcal{A}$ as $\mathcal{U}$'s agent. So when the tester, the writer, the tech supporter, the project manager, the marketer, the tinker, the tailor and the candlestickmaker all say "I Speak For The User" the immediate question has to be, "What user has appointed you as their agent?" The next question, of course is, "What about all the other users?"

I have never seen a case in which an authorized user appointed a tester as their agent.

Some users appoint testers as investigators ("get me this information") but I have not seen delegation of authority ("represent me in the meetings and make binding decisions on my behalf.")

I think that in reality, the tester has created his or her own private mental model, in good faith, of some subpopulation of the user community and is advocating consistently with that model.

- Most user communities are diverse, with diverse interests and expectations. Even if the tester has accurately modeled one segment of the community, how large and influential is that segment relative to the others? How does the tester know? How accurate are the other models (sales, marketing, project manager, etc.) with respect to the segments that they model and how large and influential are those segments? ***The key point to recognize here is that each of the participants in the triage meeting who says, "I know what the user wants and it is …" might be completely correct with respect to the segment of the user community that they have modeled. The natural feeling in this meetings ("I am right and you are wrong") is often inappropriate. The better feeling is ("which groups' interests are we reflecting and how do we get those people to reconcile in this case?")***
- When $\mathcal{U}$ is absent, unable to exercise supervisory authority over $\mathcal{A}$, we often see drift—decision-making in which $\mathcal{A}$ makes decisions believing they are in the best interests of $\mathcal{U}$, but actually over time, they gradually come to reflect what $\mathcal{A}$ wants more than what $\mathcal{U}$ wants. When I say "often", I mean this is a standard problem studied in law school courses (that most or all American law schools require) that deal with the law of agency. We study it as a routine situation, one that we will encounter time after time in different variations. Unfortunately, a mental model of a hypothetical person that you hypothetically represent is flexible. It is easy for people of good will who are trying hard to represent interests that they care about to unconsciously adjust their model in self-serving ways. There is no corrective mechanism, no reality check, that halts and reverses drift.

So, what does this mean?

Does it mean that testers should never advocate on behalf of users? *It better not, because in many development situations, users have few or no effective representatives. Even in the XP world where there is the designated One True Customer Representative, what segment(s) does s/he represent and who is being ignored? The XP process books that I have read assume that the stakeholders will work out their differences, enabling the One True Customer Representative to be* The Speaker On Behalf Of The Stakeholder. *I have not seen effective guidance in those books for achieving the often-impossible-to-achieve consensus that this assumes.*

So if testers should advocate on behalf of users, how should we do it?

1. If you are a user advocate, it is essential to come to understand the system from the users' points of view. No amount of pair programming or regression test automation can ever teach you this. Relative to the goal of user advocacy, these activities are distractions that can leave you incompetent for your task (user advocacy).
2. If you are a user advocate, don't think of yourself as *the* user advocate and don't think—unless you have definitive research behind you—that your model of the user(s) is complete or completely accurate. Rather than rejecting other views, try to work with the proponents of specifically differing views to find out what the user community(s) really believes about the disagreement.
3. All facts are friendly. Be an empiricist. In the face of argument, go get facts.