

Los Altos Workshop on Software Testing

This is a process developed by Cem Kaner and Brian Lawrence for technical information-sharing across different groups. It's not very new. We adapted it from some academic models and models from other industries. Our goal is to facilitate sharing of information in depth. We don't see enough of that happening today. Instead, what we see today is that:

- A few people read books and articles on testing. These vary in quality and there is no opportunity to ask questions of the author or challenge the data or the conclusions. We think that the BS factor in some of the "data" is pretty high and that it should be more carefully challenged.
- Some people come to conferences, where they see many short presentations. There is never time for more than a few minutes of discussion. Even in panel sessions, only a few minutes are spent by discussants on each other's opinions or data.
- Another problem at conferences is the promotion of credentials instead of the promotion of data. Someone says, "Capers Jones said X" and we all nod sagely and agree that it must be true. But who knows whether this person understood Capers correctly? Or where Capers got his data? We're mindful of an article by Bill Hetzel on software metrics in which he complained of the difficulty of tracking down the data behind many claims.
- Many people learn techniques from peers. If you stay in one company, you all work your way into a common rut. Maybe you dig out a bit by hiring a pricey consultant (aka gypsy programmer or tester). Somehow you divine from them what people in the other companies that they've worked at do. If the consultant does a good job, you steal some ideas. If not, you reject the ideas along with the consultant. This is expensive and often ineffective.

In GUI test automation, ideas that were being presented as state of the art at some conferences had been in quiet use for years in some companies or that had consistently failed or found false in practice. The level of snake oil in test automation presentations was unacceptable. Soon after Kaner started to do on-the-road consulting, he realized that many of his clients had strong solutions to isolated pieces of the test automation puzzle, but no strategy for filling in the rest. The state of the art was much higher than the conferences, but it wasn't getting across to the rest of us. Kaner hosted the first Los Altos Workshop with the goal of creating a situation that allowed for transfer of information about test automation in depth, without the BS. A meeting like this has to be well managed. As you'll see, there are no structured presentations. This is two days of probing discussion. Lawrence volunteered to facilitate the meeting. Various attendees worked for a few hours as reporters (public note-takers).

FORMAT OF THE MEETING

The workshops (we're now planning our third) are structured as follows:

1. The meetings run two days.
2. The meeting is managed by an experienced facilitator (Brian) and is recorded by volunteers.
3. We restrict the agenda to 1-3 tightly defined topics for discussion. We choose narrow enough topics that we can expect to make significant progress on at least one of them. For example, the topics in the first Workshop were:

OBJECTIVE: Develop a list of 3 or 4 practical architectural strategies for GUI-level automated testing using such tools as QA Partner & Visual Test. In particular, we want to

understand how to develop a body of test cases that meet the following criteria:

- a. If the product's user interface changes, almost no work is required to make the tests work properly with the modified program.
 - b. If the user interface language changes (e.g. English to French), little work is required to make the tests work properly.
 - c. If new features are added to the program under tests, the impact on existing tests is minimal.
 - d. A year from now, a new tester who is working on the next release of the program will be able to use these tests
 - i. with knowledge of what testing of the program is actually being done (i.e. what's actually being covered by this testing?)
 - ii. with the ability to tell whether the program has passed, failed, or punted each particular test case.
4. If there are 3 topics, we might decide that we only have enough time for two of them. For example, in the first workshop we got so interested in localization and maintenance that we deferred documentation (it's on our list for the February 1998 meeting).
5. The flow of discussion is:
- (a) war stories to provide context. Up to 5 volunteers describe a situation in which they were personally involved. The rest of us ask the storyteller questions, in order to determine what "really" happened or what details were important. Generally, stories are success stories ("we tried this and it worked because") rather than dismal failure stories, but instructive failures are welcome. No one screens the stories in advance.
 - (b) general discussion
 - (c) boil down some apparent points of agreement or lessons into short statements and then vote on each one. Discussion is allowed. The list of statements is a group deliverable, which will probably be published.

6. Publications

We agreed that any of us can publish the results as we see them. No one is the official reporter of the meeting. We agreed that any materials that are presented to the meeting or developed at the meeting could be posted to any of our web sites. If one of us writes a paper to present at the meeting, everyone else can put it up on their sites. Similarly for flipchart notes, etc. No one has exclusive control over the workshop material. We also agreed that

any publications from the meeting would list all attendees, as contributors to the ideas published. The following people have attended the first and/or second workshops: Chris Agruss (Autodesk), Tom Arnold (ST Labs), James Bach (ST Labs), Dick Bender (Richard Bender & Associates), Jim Brooks (Adobe Systems, Inc.), Elisabeth Hendrickson (Quality Tree Consulting), Doug Hoffman (Software Quality Methods), Cem Kaner (kaner.com), Brian Lawrence (Coyote Valley Software Consulting), Tom Lindemuth (Adobe Systems, Inc.), Brian Marick (Testing Foundations), Noel Nyman (Microsoft), Bret Pettichord (Unison), Drew Pritsker (Pritsker Consulting), and Melora Svoboda (Electric Communities). Organizational affiliations are given for identification purposes only. Participants' views are their own, and do not necessarily reflect the views of the companies listed.

HANDOUT: Paper published at Software Quality Week and in Software QA. An earlier version appeared in IEEE Computer.