

Invert Your Training Program: Revisited

Scott Harrigan
201 Computing Center
University at Buffalo
Buffalo, NY 14260
+1.716.645.6520
seh@buffalo.edu

ABSTRACT

At SIGUCCS 2001 in Portland, OR, I presented a paper entitled “Invert Your Training Program: Have Applicants Train Themselves Before You Hire” in which I outlined an innovative training program for new hires where we trained first and hired only those who trained the best. Although there were substantial cost and quality improvements in that process, there were still some labor-intensive aspects, such as interviewing. So, at the end of that paper, I suggested that perhaps we could completely invert the whole process, eliminating training altogether, by testing all who apply and only interviewing those who test well. This would require finding a way of testing ‘native’ or self-taught ability with computers, but would eliminate technical training altogether. And we wouldn’t waste time vetting people in interviews who we were not going to hire for low technical skill levels

In this paper, I reprise the original process to highlight the success as well as the shortcomings of that process we discovered over the years we used it. I detail how we decided to revamp this hiring process. I discuss how we made use of online, hands-on testing to permit virtually anyone to have a shot at being a computing consultant with more accurate results. I review in brief the product we used and summarize the results of this exciting new method of selecting and hiring consultants Using hands-on testing software, we have been able to expand our reach across campus to include many applicants who would not have been considered before.

Categories and Subject Descriptors: K.6.1 [Management of Computing and Information Systems] Project and People Management – *Staffing and Training*

General Terms: Management, Performance, Human Factors

Keywords: Testing, Staff, hands-on testing, Consultants, Students

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGUCCS’04, October 10-13, 2004, Baltimore, Maryland, USA.
Copyright 2004 ACM 1-58113-869-5/04/0010...\$5.00.

1. INTRODUCTION

The University at Buffalo (UB) is a large public university with nearly 25,000 students. We serve this population with public computing sites where computing and printing services are made available. We also provide trained computing consultants to assist this user community with its computing needs. The number of student assistants we employ is rather large, on the order of 110 consultants across all the sites. The hiring and training requirements for a work force this large takes up a substantial amount of staff time, and there are only 3 professional staff members, with 9 to 10 student staff members.

These limited staff resources require that we carefully manage ourselves so that we are most effective in terms of quantity and quality of the consultants we hire and the services they provide. We are constantly looking for ways to work smarter so that we have time available for managing all the other aspects of the public sites. Certainly the traditional model of hire, train, and remedially train our consultants was consuming too much of our limited resources. This led to us originally to devise the inverted training program. However, we realized even under that new regime that we were expending a substantial amount of time, money and energy that we really could not afford and this was affecting the quality of our program.

We decided that the only way of rehabilitating the program was to complete the inversion process. We had to completely invert our hiring and training process and jettison all staff intensive aspects while nevertheless improving quality. We needed a lean and mean hiring process that produced great results.

2. A RESTATEMENT OF THE PROBLEM

Until 2001, we in the User Services group would begin our training cycle that same way everyone else does: we hired the consultants first. And we hired them in a most traditional way. Applicants filled out applications that were then reviewed and interview decisions were based on that review. Performance in the interview process essentially determined the hiring decision. Offers were made to successful applicants, and they then returned for a week of training in the Fall just before school and their jobs started.

2.1 The Impetus for the Original Inversion

We would have interested people fill out an application (an online document even then) and then we would review those applications. On those applications we would have the applicant self-rate his or her technical skills in addition to the usual queries

concerning experience and references and whatnot. Of course, these ratings were subjective and largely meaningless, higher for those who overestimated their abilities and lower for those more realistic. So, in order to be sure humbleness and modesty did not cause some to overly underrate themselves, the net for interviews was cast wide. This resulted in marathon interview days, usually held on a Saturday. We often interviewed more than 80 applicants on a single day. To handle this load we had two interviews. First, a technical interview with student staff that also looked at language skills (very important in a customer service position) was conducted and if passed, this led to a customer service interview with one of three managers.

This process usually resulted in a hire or fire decision that very day based on what amounted to an informed 'first impression'. This decision had to be of poor quality because it was next to impossible to efficiently interview someone for the technical skills we were looking for due to the overall breath of knowledge and the highly graphical and intuitive nature of modern computing environments (unlike a Help Desk position which is looking for much more conceptual and, frankly, text-based skills). Even with this foreshortened technical interview, some applicants complained that they could show us the answer if they only had a computer in front of them. They had trouble articulating the answer, but they could show us their knowledge. They just couldn't seem to tell us. Nevertheless, no matter how ideal a hands-on test would be, it was too time consuming and staff intensive for us to ever seriously consider.

In any case, we wouldn't know if we had good luck in our decisions, and unfortunately luck was a big part of the equation, until we trained them several months later at the end of the summer. It was then, during the full week of training, that we would see those who were struggling. There were those who did not know much to begin with and many with no natural aptitude for computing. Nevertheless, since they were already hired by that time, we could do nothing but remedially train them. And we were not getting good results overall, so we simply accepted a lesser standard for computing competence.

So, we were not getting the results we wanted and yet it was costing us a substantial sum in a training budget to get those results we didn't want. It was a lose-lose situation for us. We needed a way to weed out those with poor technical skills. We needed to base our technical selection decisions on something more accurate than a 15-minute verbal interview.

2.2 The Original Inversion

The idea dawned on us. Why not test applicants before we hire? We could give them some training materials in advance that would reduce the training load as well. This would be a comprehensive paper and pencil exam. Such a process would allow us to better ensure that the best technically got into our hiring process, while conserving valuable staff time in the process. We would only interview those who were technically competent for the position.

This process was fairly straightforward. We took applications online as we did before, but the selection process was less stringent. We would provide electronic copies of training materials on Microsoft Office programs (a commercially purchased, site licensed product) and have one training session on

UNIX topics. Included in that training would be a micro-session on a particular Microsoft Office product to give applicants an impression of the depth of knowledge we were looking for. Applicants would then have a month to prepare for the examination. All of this would be unpaid, of course, as we have not hired the applicants yet. However, we felt there was still a valuable *quid pro quo* in that we provided the applicants with comprehensive training materials and the impetus of a test to learn the concepts well. Thus, even if they were not ultimately selected, they would have received value in learning the Office programs better.

For us, the benefits were readily apparent. This reduced our Fall paid training from 5 full days to 1.5 days. Saving 3.5 days in training and more than half of our training budget. The process should have also selected applicants for technical aptitude better than our technical interview, so that part of the interview process was dispensed with entirely. Additionally, this wider funnel should have allowed those with differing backgrounds, like less geeky English majors, a better opportunity to compete against those from sciences and engineering departments. This is especially true in that the test would not select the applicants with the best skills based on who came with the best skills, but those whose skills were best when they were tested. This leveled the playing field somewhat for international students. You could catch up if you had a drive to succeed. All the tests were directly based on and derived from the training materials and sessions, so having some computer aptitude and/or just being studious should have been sufficient to be successful on the exam. One did not have to rely on prior computing knowledge.

2.3 A Mixed Success

That was where the original inversion was supposed to take us. Did it? The answer is both 'yes' and 'no'. We did save substantial time and effort in training, not to mention the sheer expense of it all. We also eased some of the staff resource issues we had. The interview process was reduced down to a customer service interview with the manager. And we could start at the top of the list of those whose technical scores were sufficient and work our way down until we had filled the slots we needed. The test allowed these kinds of comparisons. It was also possible to select for specific skills because the exam was divided into subtests that focused on particular products or skills. For instance, performance on the UNIX subtest was a prime consideration for assignment to our Sun site.

Also, making the process competitive made the position more attractive for some. We started to draw a better kind of applicant to the position. Thus, a synergistic effect was created from making our process tougher to select better candidates: this tougher selection attracted better candidates. So the process actually made the pool better instead of just weeding the same old pool better. Some of our top candidates were even 'cherry-picked' by HelpDesk. So, the original inversion of our hiring process did a lot of what we expected it to and we deemed it to be a success from that perspective.

There were, however, unforeseen issues that blunted some of the success and caused us to retreat on other fronts. Instead of 3.5 paid training days, we now had one unpaid training day, which was optional for applicants anyway. However, that day still required significant staff time in preparation and execution. And

although we did not have to plan for 3.5 days of training each Fall and Spring, we now had a hefty paper test to design and maintain. So, our savings in time and money were not as dramatic as they might have been with a complete inversion.

Additionally, the test scores were not all that impressive, despite the amount of effort we put into designing a fair exam, and we were not able to discern where the problem lay. The test was taken right from the training material, but the questions were never normalized and so the validity of the test was a constant question in our minds. The only consolation we could take is that applicants did consistently mediocre, hiring season after hiring season. We also suspect that despite the provision of training materials, that there was very little studying actually going on. Those who did poorly on the pre-technical interview did not improve by the time they took the test. So, the hope that a determined candidate could use the process to improve him or herself never really materialized.

We also soon discovered that pre-screening for language skills was still necessary. We did not realize the full enormity of this problem right away because we were in hybrid mode for the first run of the process. We didn't trust the system so we still sent all applicants through a pre-test technical interview to get a reading on their technical level prior to taking the test. It was a check on the efficacy of the new system to produce the same or better results. This is what made the system 'hybrid'. Our system was only partially inverted. But this pre-interview also had the effect of weeding out many of those whose language was too poor to be hired, as they were unable to answer the questions properly.

The reason we still had a hybrid system with a preliminary technical examination was that we had a limitation on training by the size of the hands-on training room we had. We could only train 50 people simultaneously, and two sessions maxed us out at 100 applicants could test. As noted above, this turned out to be a blessing in disguise. We also hoped to see dramatic increases in technical ability, a training effect, but again that simply did not materialize.

Once we had a few of these hiring sessions under our belt, we realized that having the pre-test training was not showing significant improvement in scores. So, the first thing we tweaked was eliminating the training session. If training them was not significantly affecting their computing ability, then there was little point in training them. We should only test this 'native' ability. Also, if we eliminated the training session, then the only thing that limited the number of applicants was how many we could test at the same time. With the use of a cluster of large lecture rooms, over 200 applicants could be simultaneously tested. This would lead us toward a truer inversion.

We provided all the training online in documentation (we had documentation for all relevant aspects of our computing environment already). Since we no longer had any truly fixed limit in the number of candidates, we no longer had to pre-screen them for technical skills. We eliminated that part of the process. So now when an application was accepted the applicant was offered a chance to test and informed how to access the online documentation. The egalitarian system we had hoped for had arrived.

This change in interviewing opened a pit for the unsuspecting to fall into. Someone who had very poor language skills, such that we would never hire him or her, would never receive that

feedback and would continue to invest time and energy into a project doomed to failure. This person could score well on the exam, good enough to be interviewed, and yet not have a chance of being hired. I had a candidate like this myself. Even though she could apparently read English well enough to score well on a test written in it, her speaking and listening skills were so sorely lacking that conducting the interview itself was incredibly awkward. Her language skills were so poor that anyone could tell after a few minutes that she could not be hired for a customer service position no matter how good her technical skills were.

Although these cases did not substantially interfere with our process, it did make us ethically concerned that we were potentially leading people on to invest substantial time and effort in something they had no real shot at. So, we returned to pre-test screening, but shortened to just deal with language skills. These were quick 5 to 10 minute phone interviews cast as preliminary interviews but purely focused on English language ability. We did ask some confirming questions to make sure applicants understood the requirements of the position, but otherwise we were just engaging the applicant in a conversation to determine whether the applicant could understand and be understood in English satisfactorily. If they could be and had no disqualifying information, such as less than two semesters of availability, they were allowed to test.

The most interesting finding of these first few runs through the process was again that the technical interview that we had done for many years was highly predictive of how someone would do on the test. So, we could have done it either way it seems and would have had an indicator just as accurate. This suggested to us that we still had a training problem, but because we were no longer training consultants that problem was quiescent. Our training problem was simply not coming to our attention anymore.

So far, then, we were experiencing mixed results. There were definite and direct savings but we were not able to eliminate all staff intensive aspects. We had to maintain the test and the cost of delivering it. The test seemed accurate enough, but how could we be sure the test wasn't simply too hard, given that we made it up ourselves? Also there were potential issues for training itself. How do we know that anyone was learning from the process? In fact, we had some circumstantial evidence that people were not learning anything from the process, which was why we dropped pre-test training altogether.

Also, the test itself was so onerous to change, with upwards of 75 targeted multiple-choice questions, that we simply didn't do so during this whole period. This resulted in test reliability becoming more of a problem, especially as we placed no limitation on the number of times applicants could reapply. And despite all this some applicants still told us that despite their test scores they could show us how good their technical skills were if they could show us on a PC. We decided we needed something more.

3. A More Complete Solution

In the paper I gave in Portland back in 2001, I wrote: "It would seem that the most ideal situation would be to invert the process *completely* by having an 'open call' to everyone interested in becoming a consultant. An open invitation to everyone who wanted to train and take a test for a chance at an interview that they could come and be tested. Based on the results of the test, we

could then invite the best applicants to return for an interview and thus we save ourselves time not interviewing those who have no chance of making it. We would eliminate the technical interview altogether and thus consolidate and maximize the effectiveness of our time and efforts.” It would not be until 2004 that this finally became a reality.

3.1 Online Testing

It was the summer of 2002 when I first saw what would become the solution to our problem. I was assisting a Computer Science professor in charge of teaching computer literacy classes. She was having difficulties getting a program to run in one of our computer classrooms. It turned out to be a web-based hands-on simulation of Windows and Office. It was a revelation! All the questions were performance-based in a perfectly simulated environment. I took the Windows Operating System test and was amazed. The test let you answer in several ways using menus, right-clicking, or hot keys. The Word test was also astounding. And all that was needed was a browser with standard plug-ins for Flash. I realized immediately that this was how we would ideally test incoming consultants.

Online testing would solve all of our problems, as we perceived them at that time. Test maintenance would be addressed since the test makers would be updating their own product as Office changed through the years. The normalizing of questions to determine their true difficulty is the meat-and-potatoes of any testing company. So, test question creation would not only be taken out of our hands but also significantly improved beyond our ability to match. The hands-on task-oriented aspect would solve the problem that most people interact *physically* with Windows as an *environment* and therefore have trouble conveying that knowledge in a text-based paper-and-pencil exam. We would be able to cast our net even wider in order to get those who knew how to do things but were not ‘geeky’ enough to be able to conceptualize what they were doing into true/false and multiple-choice answers.

Unfortunately, at that time, the cost of these online testing systems was prohibitive. The reason was that the training/testing software only came associated with a textbook and linked to a particular individual. Each product was different and came bundled with a particular textbook. The software was not sold separately from that textbook, nor was licensing available. This made it unusable as a testing program for our purposes. We could not purchase a textbook for each and every test-taking applicant. However, I remained in contact with that professor so that I could keep abreast of new developments in the field.

In the early summer of 2003, I got a call from the professor. She was reviewing a new product called SAM (Skills Assessment Manager) from Thompson/Course Technology. The vendor was coming on campus and she invited me to attend. I was very impressed at the demo of the product. It was not so much that the product itself was particularly innovative. It was similar to a lot of the online testing products out there. What was new was how this software was being marketed. SAM was not bundled with a particular textbook. It was a stand-alone product that could be used with any number of standard textbooks. Thus, SAM got rid of the one stumbling block preventing us from using online testing: textbook bundling.

Based on that good news alone, we were prepared to fold SAM into our hiring process right away and eliminate the paper test. The first step was to figure out a price we could live with. The normal ‘street’ cost of SAM is around \$13. The bulk discount of a 250-seat license brought the price down to around the \$10.90 a seat range. This is still fairly high, but there was some flexibility on the re-use of the test if we agreed to purchase upgrades to SAM, as well as purchase individual copies of SAM and TOM (Training Online Manager) for each student we hire. This flexibility allowed us to essentially cut the per-seat cost in half.

TOM is the training software that dovetails with SAM. TOM will be folded into our hiring and training plan next year. As you might have noted, training itself had been largely dropped from the menu, so to speak. But TOM will give us the ability to customize training based on individualized SAM assessments. So, though we would still be hiring someone without training them, their test scores would be used to diagnose technical weaknesses that would generate TOM assignments to be completed within the first semester on the job. TOM will also provide a value-added aspect to the position since it tracks the Microsoft Office User Specialist (MOUS). Consultants would have the software to study for and pass MOUS certification. TOM could make the consultant opportunity even more attractive.

TOM, unlike SAM, works as a stand-alone CD, though you can link it to the online TOM for tracking purposes. This will let consultants train at home or anywhere they have access to a machine, maybe even during lulls in work shifts. We are looking forward to bringing TOM into our overall plan.

4. The Spring Hire

The hiring cycle for the Spring would be the break-in period for SAM XP. We would invite all that were interested in testing and take it from there. Since this was a web-based product, it was easy to use our own computing sites as testing stations. We only needed to load a few DLLs (Dynamic Linked Libraries) and the graphics to make the product run faster. All the rest was run from the online SAM server. We could test as many as we liked by closing down sites as needed during the slower supertime part of the day. We also changed our online application to be a bare-bones document that asked only identifying questions. We asked applicants for their name and University person number, and one vetting question: how many semesters they had left at the university. That was all. We gained their university account name from the authentication that we used to protect the page. The application also explained some of the criteria for the position including when the position started, mandatory training requirements, and having at least two more semester left at UB. In this way, we made it extremely easy to apply to be tested.

SAM was fairly easy to use when designing a test. You simply pick the skills/tasks in the subject areas you are interested in testing, such as Word, Excel, etc.. SAM puts the ready-made questions in the test. SAM even calculates the amount of time you should schedule for the test depending on the number and type of questions. Account names and passwords are easily uploaded from a file. We used generic account names so that we could reuse the accounts for the next hiring season.

In our final exam design, we had 125 different tasks that had to be completed, rather than questions. These tasks ranged from the

most simple, like cutting and pasting, to the most difficult, like inserting a watermark. This part of the exam covered Word, Excel, and PowerPoint. We created custom exam questions for our UNIX subtest. SAM does not simulate the UNIX environment, of course, but it does allow the creation of custom text-based questions. There were 18 such questions, all multiple choice. SAM only allows multiple-choice questions for custom questions. SAM does allow the questions to be presented randomly and this was selected.

SAM can be configured to make the test available only at certain times and prevent logins at all other times. SAM also individually times each test so that is a benefit for the test takers. The maximum number of tries a tester has at a task before it is marked wrong can set, up to unlimited attempts. We configured that setting for three tries. After the third incorrect attempt the test moves on to the next task.

We now had our test and we scheduled downtime in a few computer sites on two separate evenings to give it to our applicants. 269 people applied to be tested, but once we accounted for two-semester eligibility and eliminated duplicates (caused by resubmitting the simple application) only 211 were invited to the examination. Of that number, only 116 people took the exam. This was a significant no-show number to be sure, but probably reflected the ease with which one could apply and the fickle nature of many applicants.

We provided each applicant with a document that had his or her username and password for the test. The test had its own separate password that was given to the proctors to provide at the appropriate time. Everything went smoothly this first time out. The highest score on the test was 85 and the median was 62. Not overly impressive, but we took everyone who scored 60% and above for a phone interview, or 71 applicants. 45 applicants did not have sufficient technical skills for the position and were not further included in the process. After the phone interview, 64 applicants were given second interviews and 39 were offered positions.

We found this first run through to be very successful. We had results similar to our previous hires and yet expended less effort doing it. We also had a higher confidence level in the exam as a true measure of technical ability. There were, however, some problems we noted with the software. First, some applicants had trouble determining which actions were being counted as answer attempts and a few questions were lost to user mistakes. Second, although there are subtests, there is no breakout in the scoring so detailed analysis of an applicant's skill areas is not possible. This was particularly thwarting with regard to the UNIX skills subtest, which are important in determining assignments. Third, once a test was scheduled with particular 'students' it could not be altered in any way. This last fact would come to roost at our next recruitment.

5. The Fall Hire

Over the past few hiring seasons we had been trying to expand our reach to increase diversity. We have become very attractive to F-1 students who are limited to working on campus. Although F-1 international students make a good addition to our work force, there can be low-level language problems and work-hours limitation problems. Additionally, a large proportion of F-1

students are graduate students and are at a different level than the bulk of our clientele, who are mainly undergraduates, as well as short-lived in the position. Many graduate students are here only for a Master's, which takes only a year to 18 months, or they get teaching assistant or other positions with tuition waivers, which they would be foolish not to take. This leaves us with a relatively high turnover rate with graduate students. So, we are very keen on getting more undergraduates, and especially freshmen to join our cadre of consultants.

Although F-1 students have a close connection with each other and the word goes around very quickly that we are hiring, undergraduates are unorganized generally and never seem to find this out, especially when they are freshmen new to campus. We decided to target them for this hiring season with a direct mail campaign. We created a simple black-and-white 3" x 5" card insert with a picture of a consultant and just the words "We're hiring students" and the URL to our job application. We had these delivered to the mailboxes of everyone living in our undergraduate dorm. This generated loads of undergraduate traffic to our online application.

As mentioned earlier, our online application was relatively simple. However, we found at the interview stage that we didn't have enough information to get a good read on the applicant. We knew nothing about him or her and had to ask questions in the dark rather than pointed ones. So we added some questions to the application: why the applicant wanted to be a consultant; and a brief work history. In February, the application went live in conjunction with the ad campaign and stayed open for three weeks. During that period, 246 students applied for a consulting position and the process had begun.

Of the original 246 applicants, 8 applicants were rejected for the 'two semester' rule. They were graduating before May 2005. The remaining 238 applicants were offered the opportunity to take the test on one of two days. One test was to be given on a Wednesday evening and the other on a Friday evening. The test consisted of three tests. The first test was a practice exam which was not scored. This was to eliminate the problem we discovered in the Spring hire that some aspects of the SAM testing environment were awkward and unintuitive. So, the 10-question practice exam was intended to give test takers a chance learn how to take a SAM assessment exam without hurting their score. This test was optional, however.

Due to the need to separate out UNIX test scores, we had to create two separate exams. One was the hands-on graphical testing of Office products. The other was the short multiple-choice exam on the UNIX system. Applicants had to complete both tests for their scores to be considered. Applicants were allowed a maximum of two hours from start to finish.

Unfortunately, the test on the first night was not configured properly. Rather than three tries and you are out, applicants had unlimited attempts. Since the test was already scheduled, the only way to fix this was to create a new exam from a copy and reschedule. This was not possible since some applicants had already begun the test when we discovered the problem. We would have to let the test complete. Our fear was that there would be a slew of perfect scores, which would make us unable to compare test performances. Though we were very apprehensive about how valid the results of this exam would be, we were determined to keep the attempt limitation in place for the second

testing session. We hoped that we would be able to draw some useful conclusions from the first test's data nonetheless.

Again, there were a substantial number of no-shows given that there was little commitment from applicants right up to the test, other than filling out a very short online application. Based on the spring hire, we were expecting this. 197 people filled out the application, of which 178 were invited to the test. Of those offered a seat at the exam, only 129 actually sat for the test. Of the 129 test takers, 126 took both exams, validating their results, and continued on to the next step in the process. The three who did not complete both exams came from the first testing session with unlimited answer attempts. It seems very likely that they concentrated on the first exam and simply ran out of time.

Looking at the results, we found that there was not as wide difference as you might have expected between those who did and those who did not have answer attempts limited. The median score for Office exam in the Wednesday (unlimited) session was 79, not a high score given unlimited attempts. This is not even a straight 'B'. The average score for the Friday session, however, was 63. An adjustment was made of 15 points to the Wednesday people. This would make their scores comparable. One aside should be made at this point: these low scores again tended to validate our paper-and-pencil exam.

The next step was to check the applicant's language ability. This was done in a brief phone interview. Only those scoring 60% or higher were given phone interviews. That came out to 75 people. Of these, 9 applicants were rejected, and 66 people were offered personal interviews with the managers. The manager interview is a comprehensive customer service 'innerview' that basically tries to discern the applicant's genuine attitude and natural ability to communicate with people. From the whole process, we rejected about a third and hired 44 new consultants. All told, we hired 22% of the original applicant pool in the Fall hire, or just over 1 in 5.

6. Evaluation of SAM

Overall, SAM worked out exceptionally for us. It eliminated all the heavy lifting associated with testing applicants while providing higher quality results in two ways. First, all of its questions are normalized and are therefore comparable from exam to exam. We know a hard question is hard and an easy one is easy. That is very important when analyzing the results. Second, applicants have the opportunity to *show* us what they really know, where the rubber hits the road, hands on in the applications themselves. There would be little question of whether they knew or not how to perform real life tasks in the products they would be consulting on every day. There is very little that applicants can complain about concerning the fairness of the exam.

However, even though SAM does simulate the program environment very realistically, all of its simulations are focused on the question, the task at hand. There is some conservatism in the programming. Not all parts of the program are simulated in all questions (of course, HELP is never simulated). So, the parts of the program not involved in the answer are often not simulated. It is a little like those 'Choose Your Own Adventure' novels. You have choices, but only so many of them. Your choices are limited. Only certain actions work and, moreover, only certain actions count as an attempt.

For instance, the scroll bars work only if the scroll bars are part of the answer. This can lead to confusion as to why you can't scroll down when scrolling down should be, to your mind, part of the answer. It is also confusing, not to mention annoying, when the cursor has to be placed somewhere on a page for a task and only one place, the very place that is part of the answer, is made 'hot'. This is why we had to design the practice exam. Our pre-testers would become frustrated when they couldn't do things the way they wanted to and the exam would move on to the next task quite inexplicably to some. The practice exam was intended to give testers time to adjust to this.

Moreover, though, the incomplete simulation raised concerns in our minds about the possibility to 'game' the exam. Certainly, astute test takers can take these 'dead' parts of the simulation to figure out the answer. This was of special concern with the unlimited attempts problem.

We discovered this in pre-testing. I was able myself to complete tasks that I was unfamiliar with or did not know how to do. This was particularly true when the attempts were unlimited. In one task, for instance, you are to select a certain style of bullets from the dialog box, but that is the only style that is selectable, the only part of the dialog box other than the OK button that highlights with a click. Thus, it is almost impossible to get that part of the task wrong as none of the other clicks count as a wrong answer. Only if you clicked OK without selecting the bullets first would it count as an error.

Although this was as bad as it got, this incomplete simulation does cause a problem when you are trying to find out how well someone performs computing tasks, not take tests. Hence, we had a serious concern with the Wednesday session that we might not get any useful data from the test at all due to a potential for a large number of perfect scores. We were very surprised that unlimited answer attempts only accounted for a 15-point difference in the median scores.

Even with unlimited attempts, the average score on the SAM assessment was in the high 'C' range. And the test scores overall matched our previous testing experience, so this ability to game the test did not seem to be skewing things too much. The top score was 96 (although the UNIX test must be figured in). Apparently, gaming is not something we needed to be too concerned about. And to SAM's credit, most all methods for completing the task was affected in the simulation including menu-fishing, right-clicking, and hot keys.

The general functionality of SAM was very satisfactory, despite not being designed for competitive testing, but rather assessment. The website interface was intuitive, simple, and obvious. It provides many value-added features including simple reports and statistical analysis. You can also download all the testing data to do your own analysis as it imports and exports data using simple comma-delimited text files. Moreover, a single person can design and execute the exam with very little training

7. Conclusion

We have found this total inversion of our hiring process has improved things substantially. We no longer need to spend large amounts of staff time preparing for training and executing it.

Training is now limited to ‘soft skills’ needed to interact appropriately and successfully with people. We no longer have to develop a paper-and-pencil exam from scratch or find the time and money to maintain it (which we never could find). Now our test measures real life computing skills in a hands-on environment. There is now symmetry between the test and what it is testing. This way of testing technical skills should be more accurate and allows us to test anyone who is interested in testing.

This made it clear in our minds that we were not closing the doors on anyone. We have cast our net as wide as possible. And our new ‘direct mail’ advertising campaign has netted us many more undergraduates, including a large number of freshmen who may stay with us a long time cutting down on the churning of consultants.

The online exam provided by SAM was the final piece we needed to put into place to complete our inversion. And we are now poised to include TOM in the mix, which will provide us with a customized training path for those new consultants as well as a substantial perk for the position. This will bring us full circle by returning to remedial training. However, we have found a lower cost solution with SAM/TOM and this can hopefully spur self-learning when the certification potential is emphasized. SAM leveraged with TOM will provide a valuable skill set for our consultants as they leave the university. We believe that inverting our hiring process using an online hands-on assessment and training resource (like SAM/TOM) has turned a lose-lose situation for us into a win-win situation for everyone.