Authentic Testing White Paper

# Scoring Performance Tests

Wallace Judd, PhD

Performance Test Development Series 06

© 2024 Authentic Testing Corp.

Authentic Testing White Paper

# Scoring Performance Tests

Wallace Judd, PhD

Performance Test Development Series 06

# Scoring Performance Tests

Why define scoring specifically for performance tests? Isn't scoring just simply scoring?

The answer is no. Performance tests require different scoring procedures from other types of testing, as will be made clear throughout this essay.

The first reason is that knowledge may be described by at least two forms: *articulated* knowledge and *internal* knowledge.

*Articulated* knowledge is knowledge a person can explain or discuss.

Conventional multiple-choice testing evaluates articulated knowledge.

You may be able to articulate something and not be able to do it. Millions of people, after all, know the principles underlying weight loss, but are not able to put them into practice.

I was taught pole vaulting by a coach who couldn't get off the ground but was able to explain the process well enough to coach league championship vaulters.

Articulated knowledge is independent of being able to do something. It is not a prerequisite.

The false assumption underlying multiple-choice testing is that if you can answer sufficient questions to articulate knowledge, then you can in fact act on that knowledge.

*Internal* knowledge is knowledge you can act upon, but that you may or may not be able to explain.

You may know that you smell rain coming, but may not be able to explain that you smell the ozone in the air.

You may know how to do something and not be able to articulate it. When traveling over 10 miles per hour (about 16 km/h) on a motorcycle, you turn the handlebars the opposite direction from the direction you want to turn. Yet 9 out of 10 motorcycle riders will answer the other way.

If you can demonstrate that you can do something, you have the knowledge. It could be just internal knowledge and not necessarily knowledge that you can articulate.

In this essay, we are addressing only evaluations that assess the ability to do something—performance tests.

## Checking vs Scoring

In order to discuss scoring, we need to define an item and clarify a distinction rarely made—the distinction between checking and scoring.

- an **Item** is a scoreable event
- a **Check** is the evaluation of whether a response results in the desired outcome
- a **Score** integrates checks into a summary quantity

A scoreable event is anything you define it to be.

In an archery contest, you may check the location of the arrow in the target. This is a check.

Typically, the location determines the score.

But if you're teaching archery, you may score an archer on their stance, poise with the bow, drawing the bow, and release. Each of these may be a scoreable event, possibly along with the placement of the arrow in the target. Notice that what is a scoreable event in teaching archery is not a scoreable event in an archery contest.

The same is true for software or for any exam.

## Checking

Checking is determining whether the candidate fulfilled the requirements of the item.

Checks may evaluate the occurrence or the nonoccurrence of an event.

A check may look for unintended consequences of an action.

Checks may include timing.

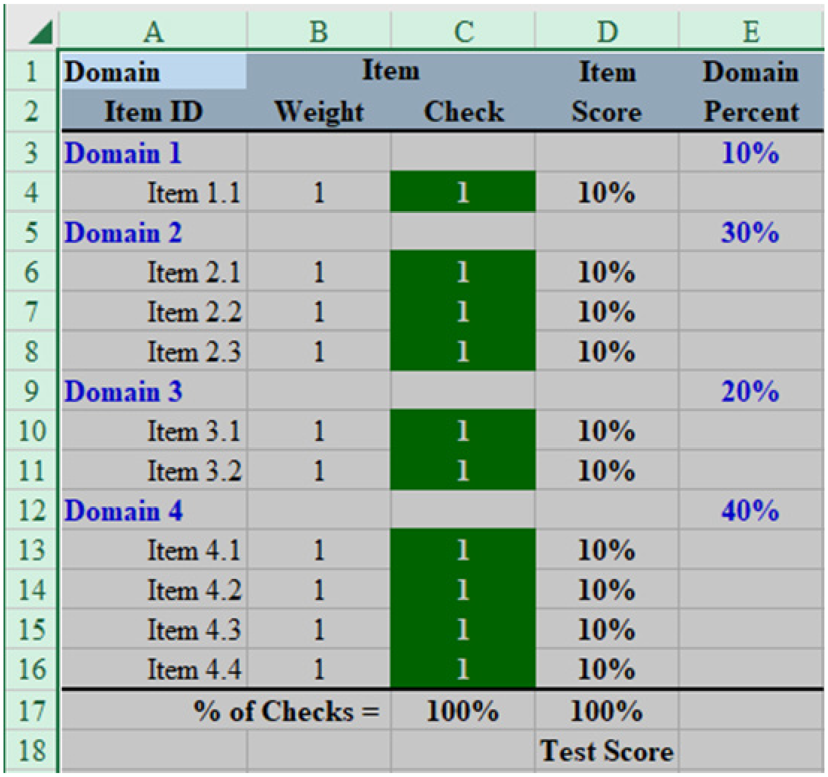
### *Timing*

Timing may be elapsed time or engaged time.

Elapsed time is time from the presentation of the item to the conclusion.

Engaged time is the time from the candidate's first action to the conclusion.

**Figure 1.** This spreadsheet shows a one-to-one correlation between the number of correct checks and the test score.



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Domain | Item | | Item | Domain |
| 2 | Item ID | Weight | Check | Score | Percent |
| 3 | Domain 1 | | | | 10% |
| 4 | Item 1.1 | 1 | 1 | 10% | |
| 5 | Domain 2 | | | | 30% |
| 6 | Item 2.1 | 1 | 1 | 10% | |
| 7 | Item 2.2 | 1 | 1 | 10% | |
| 8 | Item 2.3 | 1 | 1 | 10% | |
| 9 | Domain 3 | | | | 20% |
| 10 | Item 3.1 | 1 | 1 | 10% | |
| 11 | Item 3.2 | 1 | 1 | 10% | |
| 12 | Domain 4 | | | | 40% |
| 13 | Item 4.1 | 1 | 1 | 10% | |
| 14 | Item 4.2 | 1 | 1 | 10% | |
| 15 | Item 4.3 | 1 | 1 | 10% | |
| 16 | Item 4.4 | 1 | 1 | 10% | |
| 17 | % of Checks = | | 100% | 100% | |
| 18 | | | | Test Score | |

*Elapsed time* includes the time it takes a candidate to figure out a strategy to address the problem.

*Engaged time* is time spent on the execution of the solution to a problem after a preliminary strategy was adopted.

## Scoring

Scoring is integrating item checks into a summary quantity.

### *1 Check, 1 Point*

The simplest scoring algorithm is one check equals one point. Here is a simple example.

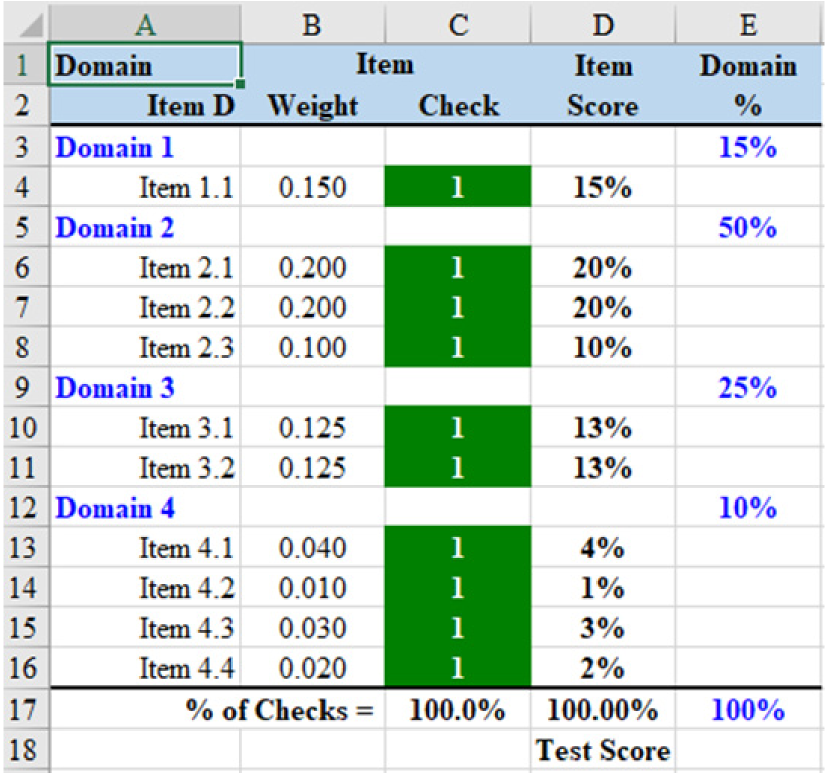
As you can see in **Figure 1**, there is a one-to-one correlation between Test Score and the percent of correct checks.

Here is this case in formulas,
Item Weight = Domain Weight / Domain Item Count
Item Score = Item Weight × Item Check

where Domain Weight is a percentage and Item Check is either 0 or 1.

Each item is worth the same percentage of the total score. And balancing the number of items needed for each domain is simple: if you are going to have 10 items, as above, just make sure there is one item in each domain for every 1/10 (10%) of score necessary for the domain. The same principle would work for 100 items, 100 Checks and a 100% total score.

This is the algorithm often used for multiple-choice testing. It's easy to work with, because if you need more weight for your domain, you simply add an item to the domain.



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Domain | Item | | Item | Domain |
| 2 | Item D | Weight | Check | Score | % |
| 3 | Domain 1 | | | | 15% |
| 4 | Item 1.1 | 0.150 | 1 | 15% | |
| 5 | Domain 2 | | | | 50% |
| 6 | Item 2.1 | 0.200 | 1 | 20% | |
| 7 | Item 2.2 | 0.200 | 1 | 20% | |
| 8 | Item 2.3 | 0.100 | 1 | 10% | |
| 9 | Domain 3 | | | | 25% |
| 10 | Item 3.1 | 0.125 | 1 | 13% | |
| 11 | Item 3.2 | 0.125 | 1 | 13% | |
| 12 | Domain 4 | | | | 10% |
| 13 | Item 4.1 | 0.040 | 1 | 4% | |
| 14 | Item 4.2 | 0.010 | 1 | 1% | |
| 15 | Item 4.3 | 0.030 | 1 | 3% | |
| 16 | Item 4.4 | 0.020 | 1 | 2% | |
| 17 | % of Checks = | | 100.0% | 100.00% | 100% |
| 18 | | | | Test Score | |

**Figure 2.** This spreadsheet shows different weights applied to each correct check.

### *Differential Weights*

The example below is just slightly more complicated. Note that the number of items in each domain is not proportional to the weight of the domain. And that the item weights are different in each domain.

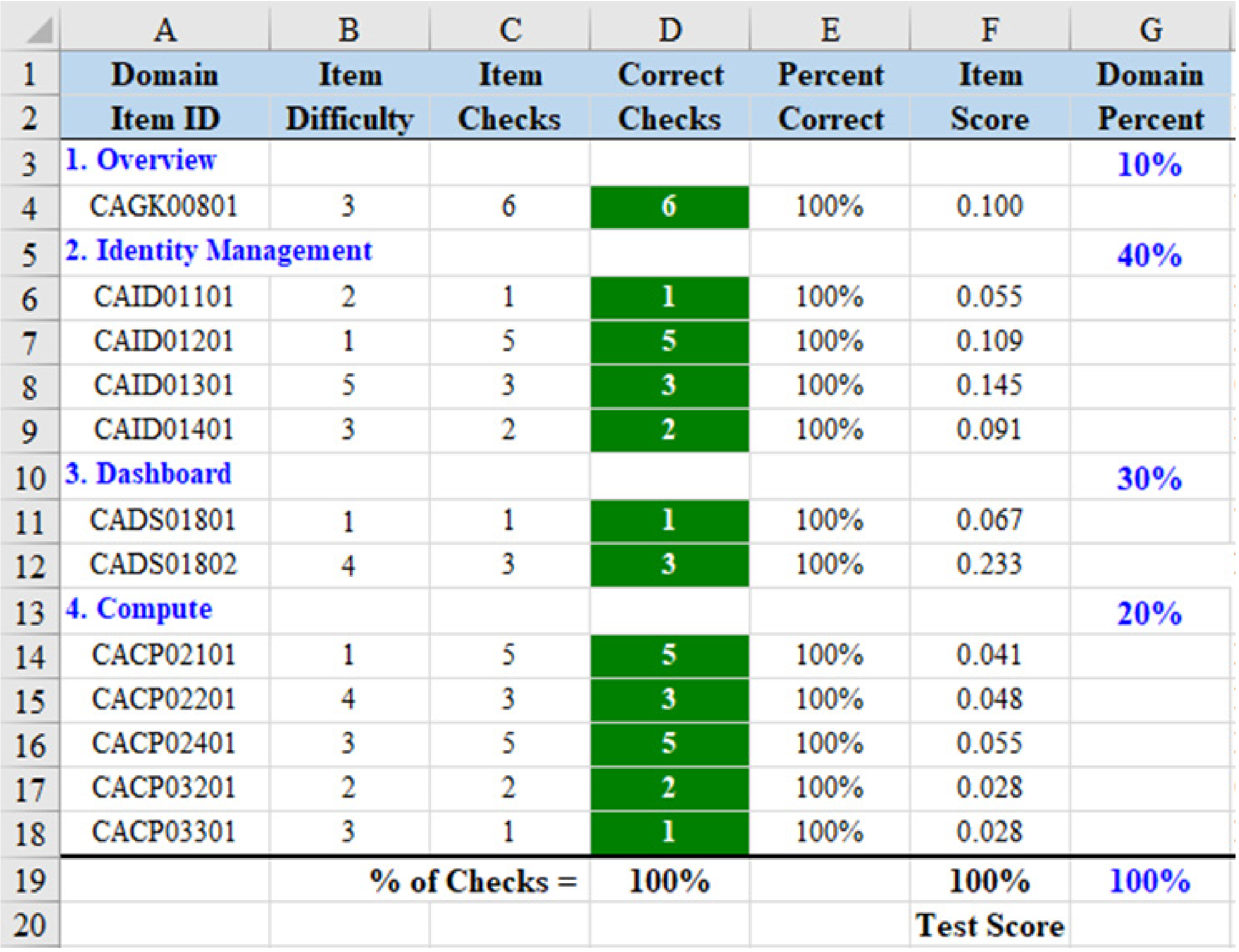
In the example in **Figure 2,** checks have different weights and contribute differentially to the total score. Item 2.1 in Domain 2 (20 percent), for example, has five times the weight of Item 4.1 in Domain 4 (4 percent).

The committee responsible for determining the item weights should have a rationale for each decision. The weight can be defined by a consensus of the committee. Or the weight could be determined by a JTA survey indicating the importance and frequency of each item. Or the weight could be determined by a regression equation evaluating the contribution of each item to a total score evaluated by a criterion external to the exam. Any of these options could reasonably rationalize item weights.

Dichotomous scoring is rational when a candidate is being evaluated on a complex task that is either fully completed or not. For example, suppose a candidate is asked to configure a specific printer to a network for a user at a specific baud rate and with a specific set of protocols. And suppose further that the candidate configures the printer, network, user and baud rate correctly, but gets the protocols wrong. The printer still doesn't work and the user can't print. If the item is instructional, it makes sense to do polytomous scoring and give the user feedback. If the item is on a certification test, it may be reasonable to assert that the printer still doesn't work and give the user no credit for configuring the printer.

## Multiple Weightings per Item

The example below is slightly more complicated than the previous one, but may best represent actual performance.



|  | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Domain | Item | Item | Correct | Percent | Item | Domain |
| 2 | Item ID | Difficulty | Checks | Checks | Correct | Score | Percent |
| 3 | 1. Overview |  |  |  |  |  | 10% |
| 4 | CAGK00801 | 3 | 6 | 6 | 100% | 0.100 |  |
| 5 | 2. Identity Management |  |  |  |  |  | 40% |
| 6 | CAID01101 | 2 | 1 | 1 | 100% | 0.055 |  |
| 7 | CAID01201 | 1 | 5 | 5 | 100% | 0.109 |  |
| 8 | CAID01301 | 5 | 3 | 3 | 100% | 0.145 |  |
| 9 | CAID01401 | 3 | 2 | 2 | 100% | 0.091 |  |
| 10 | 3. Dashboard |  |  |  |  |  | 30% |
| 11 | CADS01801 | 1 | 1 | 1 | 100% | 0.067 |  |
| 12 | CADS01802 | 4 | 3 | 3 | 100% | 0.233 |  |
| 13 | 4. Compute |  |  |  |  |  | 20% |
| 14 | CACP02101 | 1 | 5 | 5 | 100% | 0.041 |  |
| 15 | CACP02201 | 4 | 3 | 3 | 100% | 0.048 |  |
| 16 | CACP02401 | 3 | 5 | 5 | 100% | 0.055 |  |
| 17 | CACP03201 | 2 | 2 | 2 | 100% | 0.028 |  |
| 18 | CACP03301 | 3 | 1 | 1 | 100% | 0.028 |  |
| 19 |  |  | % of Checks = | 100% |  | 100% | 100% |
| 20 |  |  |  |  |  | Test Score |  |

**Figure 3.** This spreadsheet integrates multiple checks per item and item difficulty into a single score.

Domain Percent = Domain weight as determined by JTA
Item Checks = Number of checks required to evaluate the item. A measure of complexity
Item Diff = Item Difficulty as appraised by subject-matter experts (SMEs). Difficulty ratings in this instance were the SME average of 1, 3 or 5, rounded to the nearest integer. The actual difficulty scale in use ranged from 1 (easiest) to 5 (most difficult)
Checks Correct = Number of correct checks received by the candidate
Percent Correct = Checks Correct / Item Checks
Item Score = Percent Correct × Domain Percent × (Item Checks + Item Diff) / SUM(Domain Item Checks + Item Diff)

This scoring algorithm, shown in **Figure 3,** is in use in many of the tests we've developed over the past 12 years. We use it because it independently evaluates a measure of complexity (Item Checks) with a measure of item difficulty and integrates them into a single score.

## Integrating Timing into Scoring

There are tests for which timing is so critical to on-the-job success that one needs to create a score that integrates timing with successful completion of the item.

The following table was constructed to classify temporary help candidates who were performing a test involving text edits. The table integrates an Accuracy score expressed as a percentage with an elapsed

Time to complete the test in minutes.

The cells in the table were determined by managers who assigned temporary help personnel to text editing and correcting jobs.

The table in **Figure 4** is easy to read visually and clearly shows that as the time diminishes and the accuracy increases, the candidate rating improves from Fail through Basic and Intermediate to Expert.

The resulting evaluation could be duplicated with a non-linear regression equation, but it would be much more difficult to visualize.

## Gating Items

A gating item is an item that must be passed for the candidate to pass the test. Most multiple-choice developers are unfamiliar with gating items. That's because they typically don't appear on multiple-choice tests.

Gating items are addressed in detail in articles by Judd and others, but the main points will be summarized here. (See Wallace Judd, "Gating items: Definition, significance, and need for further study," *Practical Assessment, Research & Evaluation,* Vol 14, No 9.)

Gating items appear only on performance tests because they are so critical to practice in industry, and because in multiple choice testing there is a possibility of guessing correctly.

Gating items must have an unambiguous pass / fail. They are not scored polytomously.

A gating item must be so critical to the safety of the candidate's clients that to fail it would be to either endanger the candidate or the candidate's clients, or to call the candidate's professional competence into question.

In incorporating gating items, the SMEs determining the checking algorithm must achieve a consensus that the item is so critical that failing it fails the candidate.

The following are samples of gating items that demonstrate their criticality to the profession they test:

**Pilot's test.** In the FAA private pilot's checkride, if the candidate can't land the plane in three tries, they fail, despite success on any other component of the test.

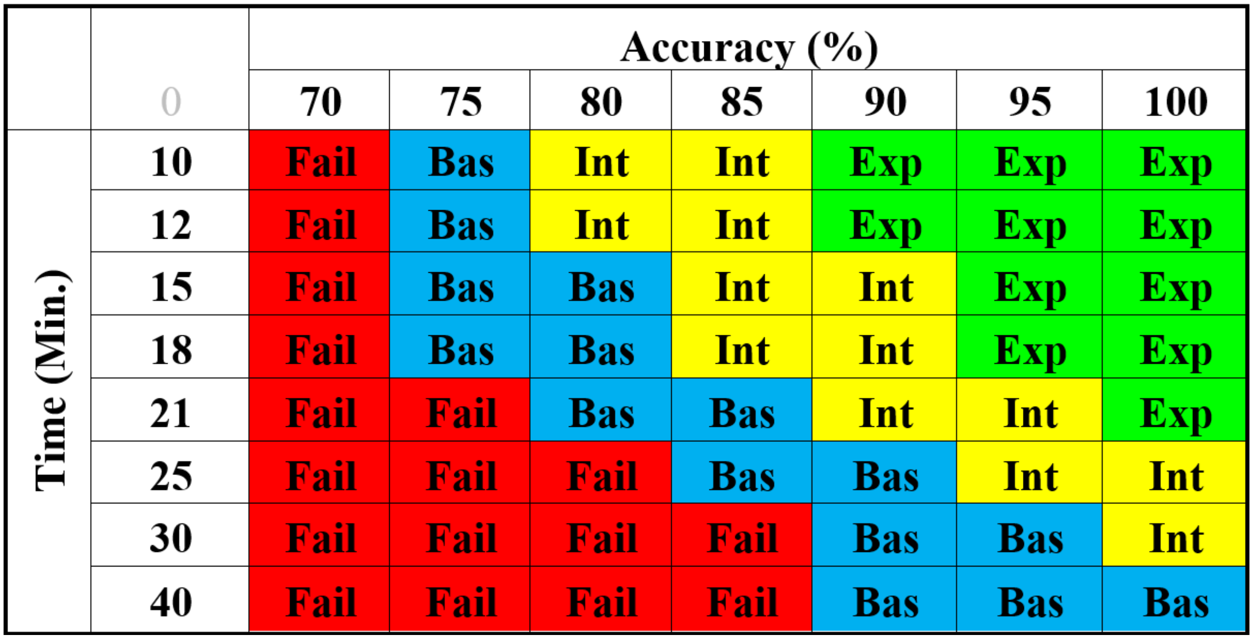
**System Admin.** In a former version of the Red Hat system administrator test, the candidate was told to fix a computer so the user could use it. If the candidate complained they didn't have the password, they were told to hack into the system and reset the user's password. If they couldn't do it, they failed the test.

**Landscaping.** In the landscaping exam, a candidate was asked to use a chainsaw to trim some shrubbery. If the candidate failed to don their safety gear before starting the chainsaw, the candidate immediately failed the test.

**Arthroscopic surgery.** In a test of vascular surgery, candidates were asked to tie off an artery they'd just repaired with an intracorporeal knot (just a knot inside the body). Candidates who couldn't tie an appropriate know with the arthroscopic apparatus failed.

**Word: Save As.** In a performance test of Microsoft Word, candidates were asked to edit a file then save the file under a new name with their initials as a prefix. Candidates who didn't know the Save As command failed the exam.

**Oracle Certified Masters.** In the Oracle Certified Masters exam, candidates were to sit for a three-day test. The first day they were asked



| | | Accuracy (%) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 70 | 75 | 80 | 85 | 90 | 95 | 100 |
| Time (Min.) | 10 | Fail | Bas | Int | Int | Exp | Exp | Exp |
| | 12 | Fail | Bas | Int | Int | Exp | Exp | Exp |
| | 15 | Fail | Bas | Bas | Int | Int | Exp | Exp |
| | 18 | Fail | Bas | Bas | Int | Int | Exp | Exp |
| | 21 | Fail | Fail | Bas | Bas | Int | Int | Exp |
| | 25 | Fail | Fail | Fail | Bas | Bas | Int | Int |
| | 30 | Fail | Fail | Fail | Fail | Bas | Bas | Int |
| | 40 | Fail | Fail | Fail | Fail | Bas | Bas | Bas |

**Figure 4.** This table integrates time and accuracy into a single performance-level descriptor: Exp (Expert), Int (Intermediate), Bas (Basic), or Fail.

to install Oracle, load some data and confirm the data definitions. The second morning they were asked to conduct some SQL transactions. At lunch after the second morning, the candidates' databases were corrupted by the instructor. When they came back from lunch, candidates said, "I can't go on. My database is corrupted." To which the instructor replied, "Just use your backup and proceed." Anyone who had failed to back up their system was sent home.

Note that in each of the instances above, failing to perform the gating item would nullify the professional practice attested to by the certification. How can a pilot be a pilot if she can't successfully land a plane? What good is a system admin who can't hack into a forgetful client's computer to fix it? A landscaper who doesn't wear safety equipment is a hazard to himself and a liability to his employer. A surgeon who can't tie off a blood vessel after an operation would let the patient bleed out. A text editor who can't save a file under a new name can only edit and replace files—never leave edits in a new version. And a database user who fails to back up the data after each transaction is irresponsible—and could be costly to her employer as well.

When I'm facilitating a session that determines a gating item is necessary, I ask that the approval be unanimous for groups of 10 and under, be unanimous except for one for groups of 20 and under, and be unanimous except for two for groups over 20.

## Adaptive Testing

Adaptive testing is an entirely different approach to assigning a value to a candidate's performance. But it is scoring nonetheless.

In classical Item Response Theory (IRT) adaptive testing assigns a candidate a location on an ability scale, formally known as the latent trait scale.

The candidate is given an initial ability assignment, then given items that are scored. The scoring algorithm assigns the candidate a new ability estimate—higher if the candidate scores correct, lower if the candidate scores incorrect. A new item is selected that yields the most information at that ability estimate. After each item, an estimate of standard error is calculated. Testing stops when the standard error of the estimate is as small as the target error.

Details of the adaptive testing algorithms are beyond the scope of this essay, but there are some informative observations which can be made.

Classical Item Response Theory rests on several assumptions:

- Unidimensional latent trait scale
- Item calibration involving a thousand items taken by a thousand candidates or more
- The ability to move from one item to any other item virtually instantaneously
- Item administration that allows many items to be administered in a relatively short time

For most performance tests, these assumptions are virtually impossible to meet.

Unidimensional latent trait scale: performance tests are inherently multi-dimensional.

Item calibration involving a thousand items taken by a thousand candidates or more: way too expensive.

The ability to move from one item to any other item virtually instantaneously: performance items can take from 5 seconds to 5 minutes just to set up.

Item administration that allows many items to be administered in a relatively short time: performance items can take from 30 seconds to 15 minutes to complete.

However, adaptive testing is feasible for performance tests. The following illustrates how it could be accomplished.

- Sort items into domains.
- Calibrate domains, and calculate inter-domain correlations.
- To administer, select the domain that has the highest sum of inter-domain correlations (Domain A), and administer and score the items in the domain.
- Select the domain that has the highest correlation with Domain A. (Let's call this domain B). Use the correlation to predict the candidate's score on domain B, and administer the item in domain B nearest that correlation. Administer items in domain B until a certainty level has been reached.
- Calculate the joint correlation of domains A and B with all other domains. Select the domain that has the highest joint correlation (C), and use the joint correlations to predict a score for domain C. Administer the item in domain C nearest this correlation, and continue in domain C until a certainty level in domain C has been reached.
- Continue in this manner until all domains have been sampled.

Authentic Testing®
The Gold Standard for Testing

3349 Monroe Avenue, Suite 336
Rochester, N.Y. 14618-5513
www.AuthenticTesting.com
+1-703-777-7321