Performance Testing Council

testing by doing®

LAB-BASED PERFORMANCE TESTING: DELIVERY AND AUTOMATED SCORING BEST PRACTICES

The Performance Testing Council, Inc.
Attention: Julie Wineberg
10309 Berkshire Rd
Bloomington, MN 55437
917-238-6228
www.performancetest.org

Performance Testing Council
testing by doing®

Created by the Design & Development Committee, Automated Scoring Task Force of The Performance Testing Council, Inc.

Acknowledgements

This document could not be possible without the expertise of the many contributors, as follows.

- Amar Rao, Biarca
- Amin Saiar, PhD, PSI Services
- Beverly van de Velde, Symantec
- BJ Dierkes, TrueAbility
- Clyde Seepersad, The Linux Foundation
- Colin Lyth, Microsoft
- Deborah Calhoun, Symantec
- Jonathan Brandt, ISACA
- Kirk Munro, Learn on Demand Systems
- Liberty Munson, Microsoft
- Marcus Robertson, TrueAbility
- Patrick Colacurio, Xtreme Consulting
- Randy Russell, Red Hat
- Tom Berry, Galileo Systems
- Wendy B. Gratereaux, TrueAbility

Design & Delivery committee chair: Ruth Ramstad

Contents

Contents

Acknowledgements 2

1. Introduction 4

Table 1: Partial Listing of Performance Information Technology (IT) Automatically Scored Labs 5

2. Historical Perspectives of Lab PTs 6

History 6

Evolution Drivers for Automated Scoring 6

3. Common Automated Scoring Challenges 7

4. Security 9

5. Scoring 10

5.1 Scoring Outcomes 10

5.2 Design Considerations 11

5.3 Scoring Within the Same Testing Environment 12

5.4 Scoring in Another System 12

6. Deployment Considerations 12

6.1 System Scalability 13

6.2 Reliability: Latency and Regional Considerations 13

6.3 System capabilities 15

7. Psychometric Validity and Reliability 15

7.1 Validity 15

7.2 Reliability 16

7.3 Psychometric Analysis 17

8. Summary 18

Figure 1: Sample IT Lab Exam Environment Overview 19

Figure 2: Sample Scoring Script: Scoring Outcomes 21

Figure 3: Sample Scoring Script: Weighted Scoring 22

1. Introduction

It is the vision of The Performance Testing Council to share experiences, knowledge, and passion in the practice of performance testing, help each other's testing efforts, and establish industry best practices. With this in mind, the Design & Development committee task force created this document to provide a technical introduction to the current state (2018-2019) of automated scoring best practices and present several critical considerations. The expectation is that the intended audience for this whitepaper is already familiar with the various performance testing design approaches. We sincerely hope you find this document useful.

Performance tests (PTs; sometimes called practical examinations) have long been used to demonstrate the application of skills. Children must pass a swimming test prior to swimming in the deep end of the pool. Drivers in most locales must take a driving test for issuance of licenses. In the US, crane operators must pass a practical exam to become certified (as of November 2018). Many medical exams include a practical component where the candidate is placed before an actor complaining of symptoms and is required to conduct the exam according to their training. Addressing the purpose of this paper, in the information technology (IT) field, some certifications require users to pass practical exams using some combination of hardware, software, and services for certification.

The advent of computer technology, and its rapidly decreasing cost, has revolutionized some PTs. For physicians, simulation products have begun to replace actors and live consultation exams. Inexpensive flight simulators that run on a home computer allow pilots to prepare for exams and certification. Particularly in the IT field, virtualization and cloud technologies have allowed for a massive – yet economical – scaling of exams that once needed expensive dedicated hardware. While the swimming test may never be replaced, cost and scale considerations are driving changes in PTs across multiple industries.

This paper focuses on "lab-based" performance assessments, which raises the question: what *is* a "lab-based" assessment?

For this paper, a lab-based assessment or exam:

- ❑ Presents candidates with the requisite hardware, software, and a series of tasks to perform, approximating the on-the-job environment very closely.
- ❑ Uses "real-world" scenarios to present tasks (aka., items[1]) typically experienced on the job.

[1] In traditional computer-based, multiple-choice testing environments, questions and answer options are referred to as "items". In performance-testing and this whitepaper, "items" are synonymous with job tasks.

- ❑ Might also describe end states to be achieved, leaving the tasks required to achieve them up to the candidate.
- ❑ Requires the candidate to use the same tools (or a high-fidelity simulation of the tools) used in the "real-world" job to solve the tasks.
- ❑ Runs in an environment expressly created for the purpose of testing **and** *not* in the "real-world" job setting.

Rather than a rigorous definition, these assessment characteristics serve to get us "on the playing field." Table 1 below provides several examples of existing lab-based testing programs that may assist in clarifying this definition.

Table 1: Partial Listing of Performance Information Technology (IT) Automatically Scored Labs

Domain	Organization	Description	Reference
IT – Applications and Operating Systems	The Linux Foundation	The Linux Foundation has a series of performance-based certification exams that cover Linux, OpenStack, Cloud Foundry and Kubernetes. These are all deployed on live virtual machines so are lab-based. This is considered a lab-based PT because the assessment does not take place in a work setting.	https://training.linuxfoundation.org/certification
IT – Infrastructure	Cisco	Cisco has long included "simlets" in its exams. These are small simulations of computer networks that are used to answer selected-response items. Additionally, for its expert-level certifications Cisco uses a combination of written and lab-based exams. These exams are scored automatically to increase the reliability of the results.	http://www.cisco.com/c/en/us/training-events/training-certifications/certifications.html Also, search for "Cisco" in the Resources section of the PTC website.
IT - Cybersecurity	ISACA	ISACA brought to market the first ever comprehensive cybersecurity performance certification program in 2015 under their Cybersecurity Nexus™ (CSX) program. Now delivered by remote proctor, the CSX Practitioner (CSXP) exam requires candidates to complete exam tasks within a multiple virtual machine environment. Scoring is initiated by candidates within their allotted exam time.	https://cybersecurity.isaca.org/csx-certifications/csx-practitioner-certification

Test sponsors face the same fundamental challenges with lab-based items as with all other item types: creating good, compelling content that measures the skills that need to be measured in a fair, valid, and consistent manner. Regardless of the exam approach, each program can add new item types, enhance the publication processes, and update policies to address common issues. Lab-based testing is unique because it can have a positive impact on candidates' perceived value of a program. Performance tests are often viewed as better representations of someone's ability and therefore more likely to distinguish qualified candidates than multiple-choice assessments. This face validity adds to the credibility of the program and adoption of the exams. Lab-based exams with high fidelity and the challenges they offer by measuring real-world skills are recognized for their reliability and credibility as well.

2. Historical Perspectives of Lab PTs

History

Performance testing (PT) has existed for centuries. Typically, the starting point is a class or course that has a dedicated practice session associated with it, in which the students use appropriate tools to apply what they have learned and to (eventually) demonstrate mastery. Within the classroom setting, an instructor is often present to offer both confirming and corrective feedback; the instructor is the assessor of the performance evaluation. Historically, these types of exams have been difficult to scale and expensive to implement. Not surprisingly, human scoring usually precedes automated scoring of PTs. However, the advent of the internet has changed performance testing, making it easier to deliver scalable, low-cost performance exams globally.

Evolution Drivers for Automated Scoring

Considerations driving the evolution to automated scoring include:

1. **Demand** – as the value of PTs is recognized, an increasing number of people take them. As a result, these volumes soon overwhelm the capabilities of human assessors; at some point, human scoring is not a scalable solution for most organizations, requiring they either limit the number of test takers, increase the number of human assessors, or automate scoring.
2. **Cost** – an increasing number of human assessors increases cost. When the cost of automating a PT is less than or equal to the combined cost of assessor training, assessor compensation, and infrastructure to support assessors, automation becomes viable (assuming the capability exists).
3. **Accuracy** – any program using human assessors is subject to the risk of inconsistency in scoring across different raters, as estimated by an inter-rater reliability (IRR) measure. Well-designed, repeatable assessments with automated scoring reduces the risk of this inconsistency.
4. **Speed of change** – APIs used for automated scoring do not change frequently, but user interfaces used to control software are changing more and more frequently, particularly with rapid software releases or cloud-based software. Automated scoring scripts often remain constant for much longer than the manual scoring steps that would otherwise be required by assessors.

3. Common Automated Scoring Challenges

Leveraging automated scoring with performance tests has many benefits, but those benefits come with inherent challenges, such as:

1. **Capability** – before one can automatically score tests, the means for automation must first exist. That is, it must be technically feasible. Three primary challenges related to capability are:
   - *Eliminating subjective performance evaluation.* Automated performance tests typically only measure what can be objectively measured. For example, it is easy to consistently measure that a cake has been baked to an internal temperature of 275°F (i.e., objective outcome). It is harder to consistently measure that it tastes good (i.e., subjective measurement). Typically, automated scoring requires limiting the evaluation to what can be objectively identified. The danger here is that limiting an assessment to exclude more subjective elements may decrease the accuracy and/or validity of the overall assessment.
   - *Constructing a scoring rubric for the assessment.* We are limited to scoring what we can measure. After defining what's important to measure, we must translate the real-world outcomes into a rubric. If one already has a rubric for a human-scored assessment, this becomes much easier to implement.
   - *Feasibility and resource limitations.* In some cases, it is not technically possible to automatically score lab resources. In other cases, when scoring is technically feasible, the logic required to score the resources may be too cost-intensive or labor-intensive to implement or may require too much time to return results (e.g., a long-running query against a database).

2. **Deciding to score the correct outcomes and/or the paths** – Many performance tests allow candidates to take varied paths and achieve intermediate outcomes in order to achieve the desired final outcomes. The automated scoring system must validate multiple possibilities for achieving a desired end-result of an exam task. However, some paths may lead to the same end-state but do so in ways that are unacceptable. For example, opening all ports in a firewall could allow mail through, *but* it would also allow malicious traffic. In these cases, item writers must describe end states to achieve with enough clarity and specificity to ensure that acceptable solutions are scored as successful completion, regardless of the path taken. And, unacceptable paths are identifiable through the requirements of the exam task (i.e., discerning unacceptable or problematic paths is often relevant domain knowledge).

3. **Candidate actions performed in the exam environment may break or impede automated scoring tasks** – All automated scoring systems have prerequisites to successfully complete the scoring of a task. However, over the course of an exam, a candidate may unintentionally, or even willfully, break these dependencies, causing automated scoring to fail. The most common of these dependencies include:

a. A scoring system that relies on prerequisites in the exam environment. (e.g., running processes on the machine, installed scoring scripts, or machine configuration).
b. A scoring system that needs access to the exam environment (e.g., system passwords, accounts used by the scoring system, or remote desktop access).

Some exams attempt to prevent candidates from modifying or deleting these dependencies by limiting the level of access or permissions that they have within the exam environment; however, some exams require the candidate to have unlimited permissions in order to properly perform the required exam tasks.

To prevent the candidate from breaking the automated scoring of the exam, exam instructions should state that modifications to the system that are not specifically required by the test may result in the exam being un-scorable, resulting in a failing score. If candidates have permissions to resources that are required to be configured a certain way for automated scoring to succeed (e.g., firewalls, network adapters) they should be warned that modification of those resources may prevent scoring.

If the candidate does break the ability to automatically score the exam, depending on exam policy, manual intervention may be required. This intervention comes in the form of modifying the exam environment so that the automated scoring operation can successfully execute. However, best practice recommends against modifying the exam environment in order to grade it, as it could result in unintended consequences and may invalidate the integrity of the environment.

4. **Output of an exam candidate's task may not leave behind significant artifacts that can be evaluated** –Test items can only be scored if some record of the candidate's actions or inputs are recorded (i.e., artifact). Examples of an artifact include a log file entry or a database entry. Items without a measurable change (e.g., Review this diagram) do not trigger the candidate to take an action that can be verified.

   To solve this challenge, item writers could:
   - Require the candidate to save an otherwise transient state[2] to a file which can then be parsed (e.g., save the output of a command to a file).
   - Require the candidate to offer a constructed response item such as a fill-in-the-blank to explain their process or show evidence of the process used.

[2]It may not be possible with an exam to score state that is transient – Depending on the timing when scoring challenges are evaluated, it may not be possible to properly evaluate the state of lab resources if those resources are changed multiple times within a lab. This may not be an issue if scoring challenges are evaluated at multiple points through a lab.

5. **People taking an automated PT may refute their score** – When automation is involved, candidates taking a PT may have doubt in the results if they receive a lower score than they expected. To avoid this problem, evidence must be gathered by the scoring system that can be used to clearly demonstrate how the score was calculated. This could include information describing the resources that were being tested, along with what was expected to receive a proper score, so that it can be reviewed after the fact if necessary. This is especially important for lab-based exams where the resources that are being scored are disposed of once scoring is complete. As an option, if costs and technology support it, the organization providing the challenge could retain disposable environments for a short period of time once scoring is complete to allow for manual review and correction of scoring results if necessary.

4. Security

The requirement to protect scoring data in storage, transit, and use cannot be overstated. Assessment programs that do not account for security and address security requirements in the earliest stages not only incur higher costs of implementation but also possess vulnerabilities ready to be exploited. Simply stated, security, both within and outside the testing environment, must be considered during the requirements-gathering phase of any project. Specifically, a secure exam program should address technical controls including:

- Protect the scoring mechanisms: Any scripts or processes involved in scoring should not be available to the test candidate.
- Ensure candidates are unable to modify the scoring mechanism.
- Make certain the evaluation mechanism provides appropriate instruction in how to complete each task.
- Leverage encryption: Test delivery should be conducted over secure channels and not open to redirection.
- Secure local storage: Test scripts and scores should be stored in a secure manner. Specific guidance to IT teams should include:
    - Secure data in transit. Test delivery and the transfer of confidential exam data and personally identifying information should only take place over secured, encrypted channels with known or authenticated end points.
    - Secure data at rest. Test delivery mechanisms, exam data and candidate personally identifying information should be stored encrypted in a secure physical location with limited access.
- Implement access control mechanisms: Only authenticated users should be able to access the exam or scoring environment as appropriate to each role.

Exam security is not exclusively a technical issue. Logistical or administrative controls are also required to improve exam security. Logistical controls can dissuade candidates from reverse engineering the scoring methodology by limiting the exposure of forms, as well as preventing exam memorization.

Logistical controls include:

- Strong candidate ID verification.
- Restricting the number of times a candidate can take an exam.
- Implementing “cool-down” periods between exam attempts.

Performance tests to be scored with automated grading solutions must be carefully designed and implemented to protect the integrity of the scoring mechanism and exam validity. Test authors may opt to implement automated grading solutions internal or external to the test environment. Within the automated scoring scripts, a test author may also implement a level of obfuscation for scoring logic. Each has benefits and shortfalls.

5. Scoring

There are several strategies for scoring exams. Platforms may run scoring scripts in place, within the same virtual environment used by the candidate. Alternatively, a platform may make copies of artifacts, or even the entire environment before remotely running scoring scripts.

In addition, while most exams complete scoring after the candidate has finished and disconnected from the testing environment, some exams score tasks as the candidate progresses through the exam.

Finally, lab activities must be scored in a way that is precise, consistent, and scalable. This section examines scoring strategies and provides best practices.

5.1 Scoring Outcomes

Complex systems may have many ways to achieve a desired state. For example, if a job task requires a user to rename a Windows computer, the user would likely do so through the user interface control panel. However, that user might also do so through the command shell, PowerShell, or even through direct editing of the computer registry. Although only a few of these methods are recommended, they can be used to achieve the desired state.

Scoring outcomes simply look to see if a user has achieved a desired state. The process, or how the user has achieved the desired state, is *not* scored by the system. In many cases, the scoring system will not be able to determine which process was used to achieve the result or will intentionally ignore any information obtained other than the final outcome of the task. This method of scoring outcomes is widespread because of its relative simplicity. Beyond simplicity and by way of example, there is likely more consensus about whether someone should know how to rename a computer than there is in the method that should be used. Furthermore, the method is highly context dependent. There are contexts in which each of the methods described might be most appropriate – again, subject to differing perspectives.

When scoring outcomes, the system scores the artifacts left by the user, making no note of the process used to generate those artifacts. It should be noted that if ‘undesirable’ methods leave any artifacts that can be automatically identified (e.g., in a log file) then it is possible to exclude those pathways from being marked as ‘correct.’ This can alternatively be thought of as assessing more than one artifact to determine the score. In this case, the script would be written so that ‘full marks IF [end state achieved] AND [undesired condition=FALSE], else zero (or a partial score).’

5.2 Design Considerations

From a design perspective, exam authors must determine:

- Should items be graded upon completion of an item (or section) or after completion of the entire PT?
- Will the task contain checkpoints?
- Must all be checkpoints be performed to receive full credit for the task or is partial credit permissible?
- Should the candidates achieve a minimum score in each section to pass or will a single score be used to determine passing status?

Scoring after test item or section allows the candidate to continue to have access to the environment but prevents a candidate from returning to tasks later. This approach is useful when exam authors are concerned later tasks might "give away" a previous item. Scoring after exam completion offers a higher level of flexibility in how a scoring engine accesses and scores tasks. The ensuing example illustrates the difference:

> In a 'score-as-you-go' environment, an exam item may require a candidate to delete row 10 of a file and subsequently edit row 25. In an 'scoring-after-completion' exam, the code required for scoring would look for the deletion of the actual text [perhaps with an 'or' statement covering lines 24 or 25] so that all 4 possible end-states can be identified: candidate gets both steps correct, candidate gets step 1 correct but step 2 incorrect, candidate gets step 1 incorrect but step 2 correct, or candidate gets both steps wrong.

Ideal scoring scripts interrogate the system (e.g., look for settings, system artifacts) without making changes to the environment as configured by the candidate. This approach maintains the integrity of the configured lab environment (useful for candidate appeals) but also allows test publishers to evaluate the impact of scoring rule updates deemed essential to remedy software bugs, patches, or other coding errors.

Policies are important and should explicitly state what actions will be measured and counted in the scoring. Additionally, test sponsors should address impacts and consequences when candidates exhibit behavior that:

- Has nothing to do with the required task
- Goes against best practice or is simply a bad practice
- Introduces a security risk
- Attempts to defeat a scoring mechanism

Security is not inherently a binary condition that exists or doesn't exist. Security is also the responsibility of the item writers to provide clarity about the specific security requirements of an item or an exam overall (i.e., they must provide a binary condition). The burden also falls to the developers when creating the exam, to place their mechanisms beyond the reach of candidates through the technical implementation, the rules governing the exam, and the text of the exam and items.

5.3 Scoring Within the Same Testing Environment

Exam authors may design their exam to be scored in real-time or near real-time within the testing environment. Tests that employ this method may send scoring scripts to a process running in a virtual environment. Because these potentially sensitive scoring scripts will be utilized in the same virtual environment to which the testing candidates have access, care must be taken to protect the integrity of these scripts and their outcomes.

Exam integrity and psychometric validity demand adequate protection against candidate manipulation of the scoring mechanism. This is particularly critical when testing highly technical audiences with advanced skills in the software. Test providers must ensure technical barriers are in place so that candidates are unable to access scoring scripts or grading logic. This is traditionally accomplished by injecting scoring mechanisms into the virtual environment only after the candidate has finished their exam and disconnected from the environment.

When scoring is in place, test providers must ensure that either:

A. The environment has been backed up. This will provide the ability for the test provider, in the case of a scoring error or a candidate appeal, to provide the original finished environment for examination.
B. The scoring scripts are non-destructive. A non-destructive scoring script is one that leaves the environment in the same state as the candidate left it. For example, a scoring script that runs a SELECT statement would be non-destructive, whereas a scoring scrupt that runs an INSERT, UPDATE, or DELETE statement would not be.

5.4 Scoring in Another System

Test sponsors may choose to grade their performance test using an external system. By doing so, they eliminate the primary risk of in-place testing where a candidate could compromise unprotected scoring scripts or logic.

Using another system, the exam environment is locked upon exam completion. Then, artifacts within the virtual environment are copied to an external scoring system for grading. This protects the integrity of the exam environment but requires maintaining the environment during the scoring process to ensure the submitted end state isn't unintentionally altered.

Scoring in a separate system introduces additional security concerns as another system must be secured. Beyond security, using a different system means that the test publisher must be able to demonstrate two additional things. First, they must be able to demonstrate that the transfer of artifacts does not modify the artifacts in any material way. Second, they must be able to demonstrate that evaluations and outcomes on the external system are consistent with the original environment.

6. Deployment Considerations

To ensure a successful lab-based testing program, test sponsors must carefully design for system scalability, reliability, and capabilities. Deployment should consider geography and latency

considerations and confirm machine configuration for consistent candidate experience. Finally, care must be taken to ensure that the environment has enough capabilities (CPU, RAM, IOPSs, etc.) for the candidates to be able to complete the tasks and do so in a reasonable time.

6.1 System Scalability

One of the main challenges test sponsors have in expanding their performance offerings is scalability. Test sponsors want to easily scale their tests to a worldwide market without significantly increasing their costs or sacrificing the fidelity of the exam. Internet based technologies have allowed some fields to scale easily; however, challenges remain. Not all exams scale easily and remotely accessed exams may impair exam fidelity. A test sponsor must carefully weigh the benefits of scale against both the cost and the fidelity of the exam.

In assessing the ability to scale an exam, test providers must ask two main questions: How do we scale the testing environment, and how do we scale the scoring of the exam?

Some PT environments are difficult to scale without significant cost or decrease in test fidelity. For example, a paper and pencil swimming test lose its value, and a driver's exam performed online is less effective than observing a driver in a vehicle who is able to recognize and avoid hazards. Other types of exams may scale easily. Many platforms exist that allow IT exams to scale worldwide with a minimal decrease in fidelity. However, even in IT, some exams are costlier to scale. Advanced exams requiring specialized and dedicated hardware may be used with scheduling systems and automated hardware/software reset scripts, but ultimately, the concurrency of these exams is limited by dedicated hardware.

Test sponsors must also consider their ability to scale the scoring of the exam. IT artifact-based scoring has been discussed elsewhere in this paper, but scoring some exams is difficult to automate. For example, data architect exams need candidates to design, but not implement, a data architecture. The exams are scored manually, by subject matter experts, who grade the candidate's design on several criteria. Because the only artifact produced by these exams is a written paper describing a design, this type of exam cannot be scaled without also scaling the number of human graders or redesigning the exam to allow for artifact-based and automated grading.

6.2 Reliability: Latency and Regional Considerations

It is important to clarify that while it is increasingly common for lab-based exams to inevitably use remote labs, it is not the only way lab exams are delivered. However, for the purposes of this section, the assumption is that lab-based exams are in the cloud (e.g., delivered remotely via a datacenter with servers, storage and applications that are connected to the Internet).

A testing program must deliver a globally consistent, high-quality experience every time. A test program can create an incredible exam experience, but latency may hurt the test-taking experience, increase test anxiety, and impact the scoring reliability. Latency, or "lag", is measured in milliseconds.

The main contributing factors to network latency include:

1) Physical distance of the transfer of information,

2) The number of hops, or networking devices, the data crosses, and

3) Computer hardware.

Latency issues can also be related to networking hardware and how the labs are set up; if the exam penetrates a firewall to connect to a remote lab, test centers may have a challenge keeping the firewall open throughout the exam. Latency could also impact exam registration if the scheduling system needs to check lab availability at the desired exam time. It can have an even larger impact of the user experience if the labs are dynamically obtained when the exam is launched. Resolutions to these types of issues can result in system delays of 15-20 minutes or more.

A test sponsor should also consider how latency affects exam timing. Establishing the exam duration depends on the complexity of job tasks and, expected latency in a real-world application of those skills. PT exam time should also include the typical expected latency that candidates are likely to experience that is a function of the exam delivery experience and beyond the real world expected latency. A best practice related to exam timing is for the exam clock to pause/stop during periods of latency that exceeds a set number of seconds. This ensures that candidates have the same amount of time to complete the task. (But it should be noted that excessive latency does in fact change the exam experience and is likely to result in a negative outcome for candidates. At the very least, they will be dissatisfied with the experience—much worse is that they fail the exam when they should have passed).

One limitation in the recommendation to stop the clock after a prescribed period of excessive latency is that it has the potential to create delivery issues. Some PTs are delivered in test centers, which creates a potential scheduling conflict if exam time exceeds what is scheduled. If remote proctors are being used, unexpected extensions to exam timing can pose issues for scheduling and availability of proctors.

Best practices to address latency for a positive test taker experience include the following:

- Explore geographical location options for proximity to test taker. (i.e., deploy close to test takers' locations to control lag by way of having a data center nearby to the test delivery location).
- Use the most effective cloud networking technology available to reduce routing over the public internet.
- Before an exam session, validate the exam workstation hardware meets the minimum hardware and network latency requirements to perform the required exam tasks.
- Consider multiple lab-based test delivery options that will ensure a good test taker experience (e.g., physical locations, remote delivery in a test taker's home or place of business)
    - Few brick and mortar physical test centers were ever envisioned to deliver modern performance exams which necessitate quality internet connections. Advances in digital pathways has made terms such as "high-speed" or "broadband" insufficient guidance to exam candidates. Instead, exam sponsors should establish minimal "up" and "down" internet speeds that considers their program needs, user experience, lab locations, and geographic limitations.
    - With regards to remote delivery, exam sponsors should offer a compatibility check mechanism that candidates can run to verify the computer they will test on meets minimum specifications (i.e. processor, memory, browser plug-in's, firewall settings,

etc.). This is especially important at businesses which invariably have strict security configurations for specific business needs.
- Regardless of mechanism, support processes should exist and provide clear direction to resolve common issues.

6.3 System capabilities

During development and quality assurance (QA), the actual candidate experience is often overlooked. Some reasons for this include: the development/QA environment has better resources (RAM, CPU, IOPs, etc.) than the candidate will have; development tends to look at each task in isolation while candidates create a cumulative load on the system over the course of the entire exam; candidate errors may create load on the system that development doesn't experience; or, the development environment may be co-located with the terminals resulting in artificially low system latency during development and testing.

The 'cure' for this is, of course, is to do early and frequent 'real world' testing where candidates (ideally spread around the world) do alpha testing of tasks both individually and in sets. This iterative approach is tremendously helpful in identifying potential system constraints that can be addressed by increasing the specs for the candidate environment. It should be noted, however, that increased specs leads to increased cost to operate the program so there may be instances where the 'right' solution is to instead re-engineer a task so as not to require more powerful (and expensive) candidate environments.

7. Psychometric Validity and Reliability

7.1 Validity

Psychometrically, PTs are treated much the same as more traditional exams. To ensure validity of the examination content, test providers should ensure that the tasks are built on the competencies (i.e., knowledge, skills, and/or abilities) identified through a job analysis (JA). Prior to the JA, the test provider should have given some thought to the structure of the exam. Will it only contain performance items, or will other item types be available to cover the competencies that cannot be easily tested with performance items (e.g., those tasks that take excessive amounts of time to run/process in the real world)? Either way, the inclusion of performance items must be clearly communicated to subject matter experts participating in the JA because this may influence the types of competencies that they identify during the process. This may increase the complexity of the competencies identified and will likely cause the SMEs to consider competencies that are important but may not be realistically tested with performance items. Similarly, any delivery and scoring limitations should be shared with the SMEs because this will ensure that only testable competencies are identified through this process.

The competencies identified through this process should be evaluated in much the same way as they are for traditional exams (e.g., importance, frequency, point of acquisition) and should result in an examination content outline (aka test blueprint). The resultant blueprint should be used as a guide for item writers (i.e., task developers), informing them of the number of items required to assess each competency.

Item writers should be directed as to the scoring processes used for items so that they can construct items that align with the requirements for each scoring process. A few important considerations exist here:

- **Variable weighting vs. Uniform Weighting** – Generally, allowing different items to have different "point values" is not recommended. Instead, the common and recommended approach is to allow for repeated measure of a given competency. If repeated measures are not feasible, then assigning point values according to some identifiable data is recommended, such as ratings of importance derived from the job analysis process.
- **One Multi-Step Item vs. Several Separate Items** – Items can refer to how the information is displayed and/or how the information is scored. If multiple steps are required to perform a task described in an item and they are highly inter-related or inter-connected, then it may provide support for including them together as one single item. If the item can be broken into separate components and each step can be assessed separately and provide a meaningful measure, then there may be support for splitting them up.
- **Dichotomous vs. Polytomous Scoring Models** – Dichotomous scoring refers to all-or-none scoring. Polytomous scoring refers to partial-credit. An example of a dichotomously scored item is one in which the candidate must include all the necessary elements in order to achieve the point, whereas a polytomously scored is one in which a candidate gets a portion of the total point value based on each correct element. Polytomous scoring usually requires tracking "checkpoints" in the task so that candidates can be awarded credit for the parts that they did correctly rather than requiring them to perform everything correctly. A key benefit of polytomous scoring is that it can be used to assess multi-step items with increased accuracy but may require a more complex overall scoring model.

As with traditional assessments, a different set of SMEs from those who wrote the items should review each item to ensure the accuracy of the content and its link to the blueprint. If there is a discrepancy, the task should be discussed, and consensus reached on the competency being measured. The exam should then be assembled to include tasks in the appropriate proportions to map to the blueprint requirements.

If possible, showing that those who have passed the PT are better performers on the job is an additional way to demonstrate the validity (i.e., criterion validity) of the exam. This process establishes the relationship between the assessment and some outside criterion of importance (e.g., job performance ratings, on-the-job productivity measures). That said, the availability and accuracy of sufficient data related to the criteria are common challenges associated with evaluating this.

7.2 Reliability

Calculating reliability continues to be a challenge for PTs. If the PT covers a small subset of highly related tasks, then the typical reliability indices, such as Cronbach's alpha or the KR-20, are appropriate. However, the very nature of PTs allows test providers to evaluate a broader domain of competency. As

the tasks become more diverse, these reliability indices are less appropriate and likely underestimate the reliability of the exam. Adding polytomously scored tasks exacerbates this effect.

Another approach to reliability that may be appropriate is test-retest reliability, but it is less often used to evaluate assessments. This type of reliability is the correlation of the scores between multiple attempts at the exam. Given the challenges with alpha and KR-20 for PTs with a small number of items or a diversity of content assessed, this may be a better evaluation of the reliability of a PT. However, test-retest reliability also has limitations, especially if the item bank contains a limited number of items, meaning that candidates who retake the exam are likely to see some or many of the same tasks because the repeatability of tasks will likely inflate scores across time, resulting in lower estimates of reliability.

A more creative approach to reliability might be to compare scores of "equivalent" groups of candidates – or even "equivalent" candidates. Equivalent groups or individuals should earn similar scores; thus, correlations between these groups or individuals could provide a reasonable estimate of reliability. Of course, defining 'equivalence' is the central challenge with this approach to reliability.

Psychometricians are increasingly questioning the appropriateness of traditional approaches to reliability for PT as they find more instances where they may not be appropriate for PTs, especially those that include polytomously scored items. Until more research is done in this area, psychometricians will very likely continue to use the traditional indices with the understanding that they most likely represent the lower bound of the reliability of exam.

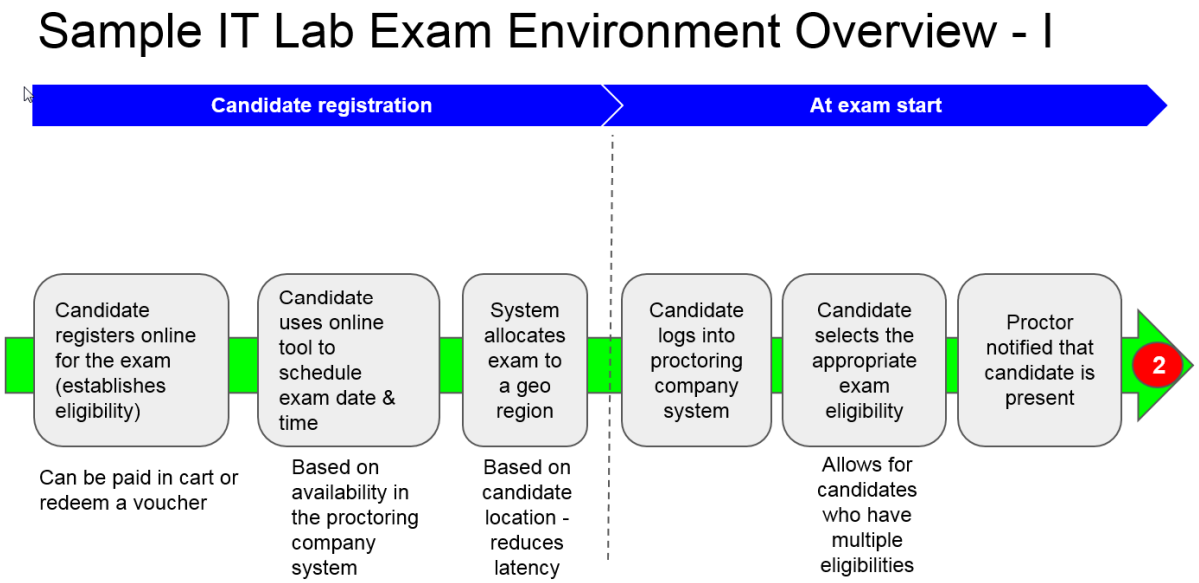
7.3 Psychometric Analysis

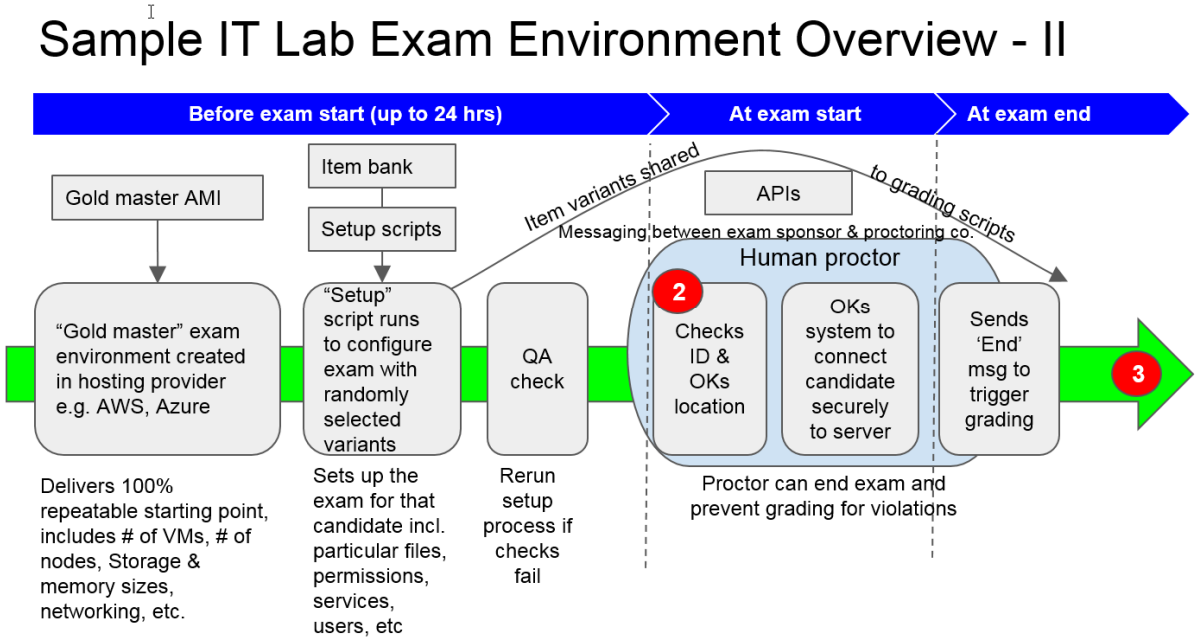
Classical test theory and item response theory statistics are calculated in the same way for performance items as they are for traditional exams. The key difference in the psychometric analysis of performance items is in the option analysis. Obviously, performance items don't have defined options. However, if they have checkpoints (which are strongly recommended for reasons that will become apparent shortly) that are tracked (even if they are not scored separately), these checkpoints can be analyzed in much the same way that options are. In fact, one best practice is to review the difficulty (percent of people performing correctly) of each checkpoint and its differentiation (point biserial correlation between performing that checkpoint and overall score on the test or task) of candidates to understand if the checkpoint itself has psychometric issues. This type of analysis is especially helpful if a task is not performing well psychometrically. For example, an item may not be performing well because of a single checkpoint that can be removed or modified much more easily than recreating the entire task. This analysis allows the test provider to fix issues at the checkpoint level that may "save" a task that may otherwise be lost because of poor psychometric performance. With this said, checkpoints might aid psychometric analysis, but they also presume a specific path and process or possibly a small set of known paths and processes. Depending on what is being tested, checkpoints might be safe and desirable or might be something that should be avoided.

8. Summary

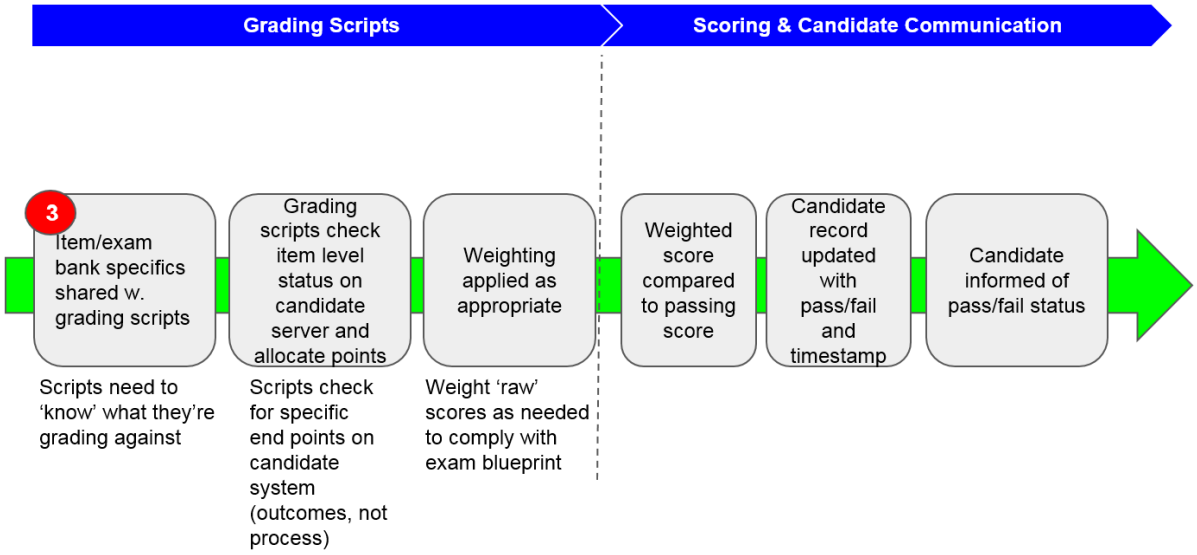
The certification industry has evolved beyond traditional multiple-choice testing into new ways of validating skills through performance testing. Global organizations realize the importance of scalability, particularly for technical performance testing. Automated lab exam scoring offers organizations the capability to scale without taking time away from valuable subject matter experts to manually score lab exams, regardless of whether the exam was delivered in a physical or virtualized environment. Our intention was to increase understanding of performance tests, identify challenges, and demonstrate industry best practices for automating lab scoring which benefits a broad audience. If nothing else, we sincerely hope this paper sparks conversation as to whether a performance test is a feasible modality for your particular assessment requirement.

Figure 1: Sample IT Lab Exam Environment Overview





Sample IT Lab Exam Environment Overview - III



<Intentionally left blank. See next page.>

Figure 2: Sample Scoring Script: *Scoring Outcomes*

The sample scoring script is developed to measure if a candidate successfully created a virtual machine (VM) of a particular size and operating system (OS) in Azure.

Candidate Job Task Instructions:

Create a new Windows Server 2012 R2 Datacenter virtual machine named Server01 of size Basic.

```
///
/// EXCERCISE:   <01 - Create a new Windows Server 2012 R2 Datacenter virtual machine>
///
task1 =
{
    // The id, name, and description of the task.
    // There may be multiple tasks checked in a script
    var id = 'L2AE01T01'
    var name = 'Deploy WS2012R2 Basic_A0 Datacenter VM'
    var description = 'Create a new Windows Server 2012 R2 Datacenter virtual machine
named Server01 of size Basic'
    var success;
    var message;
//  Psuedo-code for the check itself
    var check =
    {
        var vm = GetVm('Server01')
        //  Check whether the vm exists
        IfExists(vm)
    {
        // Check the vm size and OS
        IfSizeIsBasic(vm) AND IfOsIs2012(vm)
        {
            success = true;
            message = 'Server01 is a Basic_A0 Windows Server 2012 R2 Datacenter Azure
VM';
        }
        else
        {
            success = false;
            message = 'Server01 it is not running Windows Server 2012 R2 Datacenter or is
not of Basic_A0 size';
        }
    }
    else
    {
        success = false;
        message = 'Server01 does not exist'
    }
}
```

Figure 3: Sample Scoring Script: *Weighted Scoring*

Figure 2 shows a basic script, however, it returns a simple true/false and a message for whether the user completed the task correctly. Another option involves weighting various artifacts scored within the task, and returning a decimal score.

To maintain compatibility with learning management systems, this sample scoring script presumes the value returned for scored tasks is typically a decimal between zero and one.

In the script below, we score the *same task*, but we provide a more complete audit trail by returning an array of messages, and weight the various artifacts scored. In this example, creating the virtual machine (a gating task) is worth half of the total score, and creating a virtual machine with the correct size and operating system are each worth a quarter of the total score.

```
///
/// EXCERCISE:   The same with multiple tasks with weighted scoring and a more complete
audit trail
/// Pseudocode
///
task1 =
    {
        // The id, name, and description of the task.
        // There may be multiple tasks checked in a script
        var id = 'L2AE01T01'
        var name = 'Deploy WS2012R2 Basic_A0 Datacenter VM'
        var description = 'Create a new Windows Server 2012 R2 Datacenter virtual machine
named Server01 of size Basic'
        var weightedScore = 0;
        var messages[];
        //  Psuedo-code for the check itself
        var check =
        {
            var vm = GetVm('Server01')
                //  Check whether the vm exists
                //  This is a gating item and worth half of the total task score
            IfExists(vm)
            {
                weightedScore = weightedScore + .5;
                // Check the vm size, worth .25 to score
                IfSizeIsBasic(vm)
                {
                    weightedScore += .25;
                    messages.Add('Server01 is a Basic_A0 Windows Server.');
                }
                else
                {
                messages.Add('Server01 exists, but the size is {vm.Size}');
                }

                if(IfOsIs2012(vm)) {
                    weightedScore += .25;
                    messages.Add('Server01 is a Windows Server 2012 R2 Datacenter Azure
VM');
```

```
                }
                else
                {
                success = false;
                messages.Add('Server01 it is not running Windows Server 2012 R2
Datacenter.  The OS is {vm.OS}.');
                }
            }
            else
            {
                messages.Add('Server01 does not exist');
            }
        }
    }
```