THE LIFE CYCLE AND QUALITY ASSURANCE OF PERFORMANCE ASSESSMENT BATTERIES

Robert E. Schlegel and Kirby Gilliland
Center for the Study of Human Operator Performance
University of Oklahoma
Norman, OK 73019

ABSTRACT

As software products, computer-based performance assessment tasks and batteries cannot escape one of the cornerstones of software engineering – the software life cycle. This paper presents a discussion of the elements of the software life cycle that are unique to performance assessment batteries and focuses on a specific element of product development, quality assurance assessment. A discussion of the key ingredients for converting a computer-based assessment task into a commercially viable product is also included.

INTRODUCTION

As software products, computer-based performance assessment tasks and batteries cannot escape one of the cornerstones of software engineering – the software life cycle. One representation of the software life cycle incorporates the following activities: requirements specification/analysis, architectural design/ specification, detailed design/specification, coding and unit/module testing, integration and testing, and operation and maintenance. All activities except operation and maintenance are typically considered part of the development process. Operation and maintenance activities, which occur after product release, feed back to all other activities to impact future product development. Maintenance continues until a new version of the product demands a total redesign or the product is phased out. This cycle has been observed several times in the history of performance assessment batteries (PAB), primarily driven by technology advances from DEC VAX and PDP minicomputers to Apple II and Commodore 64 microcomputers to personal microcomputers running DOS and then Windows to Personal Digital Assistants.

In their relatively short twenty-year history, PAB's have evolved from converted paper-and-pencil clinical assessment instruments and computerized versions of older-generation electromechanical laboratory performance tasks to innovative, interactive tasks and simulated micro-worlds that can engage participants. Modern PAB's have released us from pencil and paper or cumbersome electromechanical devices, have lowered the cost of testing in many ways, have allowed us to make more accurate and elegant versions of tasks, and have extended the limits of our assessment potential. At the same time, there are a number of legacy issues associated with the expedient reuse of old, sometimes inefficient, computer code and attempts to retain the validity demonstrated in older test forms. Some tasks have been computerized simply because conversion was feasible, resulting in tasks that may no longer be relevant or valid. In other cases, good tasks have been poorly implemented, resulting in tasks that are neither valid, sensitive, diagnostic, nor accurate.

Life Cycle of a Performance Assessment Battery

One representation of the software life cycle of a performance assessment battery would include the following elements:

- PAB Originating Circumstances: Theory or Problem-Based
- General Task Requirements and Specifications
- Detailed Software Development and Battery Testing and Evaluation
- Psychometric Properties, Validity, Sensitivity
- Factors Affecting Task Battery Utilization
- Funding for Development and Support

- Marketing, Promotion, and Sales of PABs and Services.

In many settings, all activities except the last two (future funding and marketing, etc.) are typically considered part of the technical development process. Operation and maintenance activities, which occur after product release, typically feed back to all other activities to influence future product development. Maintenance and/or product services usually continue until a new version of the product demands a total redesign or the product is phased out.

The Balanced Performance Assessment Battery

Figure 1 presents the key elements proposed by the authors for the successful commercialization of a performance assessment battery. The principal foundations are innovation, development, evaluation, and application, followed closely by a solid underpinning of customer support. The fundamental element for success is a good idea in terms of task conceptualization that correctly taps the skill or resource of interest and takes advantage of state-of-the-art technology. The idea must then be fleshed out in terms of task specifications, module programming, and packaging. Evaluation of the software is critical to demonstrate to the user community that the task is an accurate and valid assessment instrument. The following section on quality assurance (QA) elaborates on this topic. Determining likely applications of the task or battery provides the impetus for appropriate marketing of the software. The final critical element is customer support in terms of services such as a help line, bulletin board, website, and specialized support.

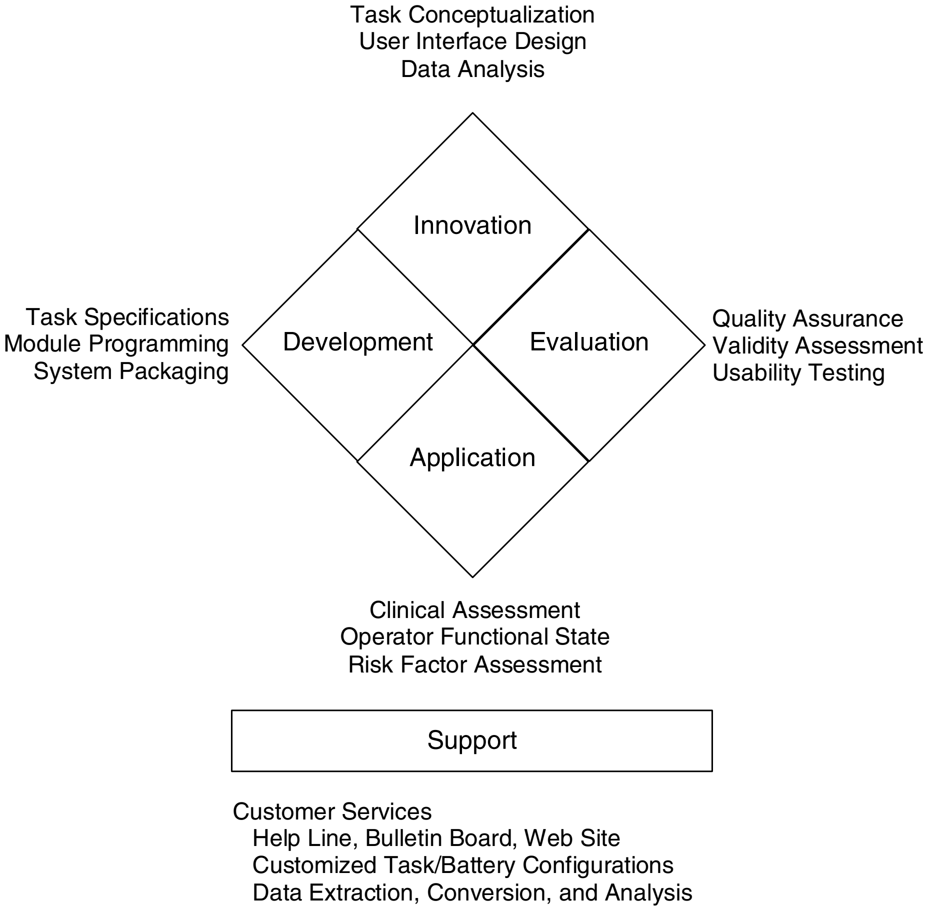

Figure 1. The Balanced Performance Assessment Battery.

QUALITY ASSURANCE ASSESSMENT

Few PAB's undergo extensive QA review before distribution. However, PAB’s vary greatly in the sophistication and quality of the rendering of their component tasks. Years after their introduction and use in research, we often find fundamental programming or implementation problems that could have serious implications for the data that have been collected. A QA evaluation should represent a required initial step before further scientific evaluations addressing psychometric properties and validity can be performed.

The QA review of any product can be a complex and time-consuming process. The QA assessment of software is additionally complicated due to the myriad number of logic branches that must be validated and the number of configurable parameters that together produce a seemingly limitless number of potential test conditions. In some cases, the review of a simple task may be completed by a single individual in a day or two. More commonly, the review involves a multi-person team and multiple days or weeks.

As in the inspection of a physical product with numerous features, it is often advantageous to train inspectors to become specialists in searching for a specific subset of quality characteristics. This approach strengthens the expertise of the individual inspectors and increases the likelihood that flaws or defects are identified. Utilizing a cadre of inspectors also shortens the time required for completing the lengthy QA assessment.

An important first step in the QA assessment of a battery is understanding the origin and intent of the PAB developer. Each PAB is usually designed for a purpose and often has theoretical roots that extend or limit its applicability. Understanding this aspect of a PAB is important in evaluating and understanding the nature of the PAB.

The next critical step is actually acquiring and installing the battery. The acquisition of some PAB’s is not an incidental task. One often has to locate the PAB developer, acquire the software (while understanding that diskette, CD-ROM, FTP transfer, etc. are not always intuitive processes), and negotiate any intellectual property rights or disclosure agreements. This negotiation can involve institutional approval and in some cases require legal opinions on one or both sides of the transfer.

Installation may also be a difficult and time-consuming process. Rarely is PAB documentation clear, concise, and thorough. Editing installation instructions and other documentation, such as user’s manuals, is often among the lowest priorities of PAB developers. Therefore, it is important to acquire any instructions that are available and evaluate them for completeness and accuracy. Documentation for previous versions of software provides an expedient foundation for new editions, but it is very easy to overlook important changes in design or operation in editing the manual. One way to discover such problems is to read instructions and follow them in naïve fashion. In addition, it is important in the QA process to document all difficulties such as missing files or missing stages in the installation process.

The next step is defining the scope of the QA assessment. This involves listing all tasks and task parameters, the range of the parameters to be tested, characteristics of the stimuli (structure, appearance, generation rules, sequences, etc.), event timing procedures, user feedback, and any other characteristics of the PAB. While there are an almost infinite number of factors to evaluate in a QA assessment, several elements seem to be of major concern. The following list provides a brief enumeration of the most important factors of general concern:

- Instructions to experimenters
- Task instructions to users
- Task parameters (range of values, default values)
- Test stimulus visual and structural characteristics (stimulus generation rules)
- Test stimulus sequence characteristics (stimulus generation rules)
- Timing precision and accuracy for task events and responses
- Response recording procedures and metrics
- Operating shell characteristics.

The QA evaluation should also address software usability from the standpoint of the test participant and the examiner or experimenter. It is easy to overlook this dimension of a PAB. One of the most common problems we have encountered in PAB evaluations and use is knowing how to stop the battery in action and how to restart it. This is essential in the QA process unless you have a lot of time on your hands to wait while the tests timeout! Surprisingly, while escape keyboard functions are almost always available in PABs (if nothing more, they were essential in the programming process), few PABs provide such fundamental information, let alone more detailed information.

The presentation of the results of a QA assessment can be an important matter in itself. Long narrative reports can be useful to designers and researchers who may be interested in the underlying issues in test construction. However, it is likely that programmers may get greater use from a systematic list of issues sorted by task (or by category across tasks) and identified by severity of issue. Consideration of providing a sortable spreadsheet summary of problematic findings as illustrated in Table 1 is strongly advised.

Table 1. Example of Sortable Software Issue (“bug”) Report.

Bug #	Resolved?	Category	Priority	Module	Version	Problem Description
1	OK/nc	Module Information - Version Control	8	MTH	1.31	Task Naming: "Math Processing" title bar and Demo list, "Mathematical Processing" (elsewhere).
2	OK/nc	Module Information - Version Control	8	MAT	1.31	Task Naming: "to" is capitalized in database and title bar, but not in Demo list
3	OK/nc	Module Information - Version Control	8	RT	1.31	Task Naming: "Simple RT" (title bar), "Simple Reaction Time" (elsewhere). Also, "reaction" not capitalized in Demo list.
7	Y	Initial Title Screen	7	SLP	1.31	Menu Bar Activation: no drop-down hotspot.
9	Y	User Instructions	8	SLP	1.31	Stimulus description: words could be moved up on lines to fill white space.
15	Y/N	User Instructions	6	SLP	1.31	Key map: unused keys not grayed out in the same fashion as other tests.
23	Y/N	Warmup Trials	5	MTH	1.31	Warmup feedback: after third warmup series failure, no feedback on number missed.
24	Y/N	Warmup Trials	5	RMM	1.31	Warmup feedback: no feedback given for last trial if warmup is passed.
40	Y	Abort Handling	2	MAT	1.31	When the PC is powered back on after reaching task idle deadline, battery continues at the task AFTER the task that was running…
43	Y	Trials/Feedback	4	MAT	1.31	Number of Trials: 14 in battery, 15 in Demo. {now all 15.}
45	OK	Trials/Feedback	4	MEM	1.31	Number of Trials: 40 in tech manual, but 36 in Demo due to new stimulus generation rules.
52	?	Data Accuracy	3	All		As a result of the definition of Percent Correct (Cor/(Cor+Err)), a single correct response followed by lapses until task timeout results in a feedback of 100% Correct.
92	Y/N	User Instructions	8	VIG	1.01	Instructions: missing period at end of first sentence.
94	Y/N	User Instructions	8	PMS	1.01	Instructions: text should be left-justified, not centered (on second screen).
104	OK	Response Configuration	6	VIG	1.01	Start key: key 5 is used rather than key 3 (which is used on all other tasks).

SUMMARY

Based on the authors’ expertise in task battery development, evaluation, validation, and application developed over the past twenty years, this paper has presented a summary of the life cycle of a computerized performance assessment task/battery, the key ingredients for turning a computer-based assessment task into a commercially viable product, and the critical elements of a quality assurance assessment. Users of cognitive performance tasks and batteries, whether clinical practitioners or experimental researchers, should be aware of the limitations of the products they use.

The purpose of this article was to provide an opening dialogue on quality assurance (QA) in PAB design and development, and to ultimately improve the development, evaluation, maintenance, and application of PABs. Due to the broad scope of such an endeavor, many of the statements in this document were generalizations based on the authors’ two-decade history of work in computer-based assessment. While exceptions to the generalizations may be plentiful, the statements were intended to demonstrate the numerous benefits associated with the application of even a modest level of QA in PAB development or *post hoc* evaluation.

It is hoped that the presentation of this information will empower consumers to purchase and use such tasks and batteries with a better understanding of the basic characteristics that make the tasks valid and useful. In addition to considering the suitability of a task for the given application and the acquisition cost, it is essential for users to demand a documented warranty of the accuracy, reliability, and validity of the task software implementation.