TDS Feature Evaluation

Prepared for: SMARTER BALANCED

Fairway Technologies, Inc.
April 27, 2016

Table of Contents

Executive Summary

Fairway reviewed the original proposal and each of the seven amendments in order to determine the requirements for the final delivery of the Test Delivery System (TDS). We categorized each of the 78 requirements as fully implemented, partially implemented, not implemented or unknown based on the information available. It is important to note that Fairway did not have access to emails, meeting notes or other communications where decisions might have affected the deliverables. We based our evaluation on the original RFP, contract, proposal and amendments.

Overall, two requirements have not been implemented, 56 were implemented as expected and the remaining 18 were implemented to varying degrees.

Overview

Process

Fairway was asked to evaluate the Test Delivery System (TDS) features to determine if the original requirements were fully implemented. Fairway reviewed the original Smarter Balanced RFP, AIR's original proposal, the seven amendments to the original contract and the TDS application code as of December 2015. We deployed the TDS application and necessary shared components to an environment in AWS in order to evaluate the features.

Fairway reviewed requirements that were specific to the Test Delivery System, which for the purposes of this evaluation include the Student and Proctor applications, Test Integration System (TIS) and Teacher Hand Scoring System (THSS). The single sign-on component was also reviewed in the context of providing a secure environment across the components within TDS.

The RFP and proposal defined 130 requirements which Fairway reviewed and determined 68 of those requirements are related to TDS.

Summary

Fairway evaluated each requirement related to the Test Delivery System as defined in the original RFP while taking into account any changes introduced in the amendments. The table below summarizes our findings. Each item evaluated has been categorized as to whether it was fully implemented or not. A value of "Yes", means fully implemented, "No", means that it was not implemented, while "Partial" means there are gaps in implementation or differences from the original requirement. Additionally, "Satisfied" is used in a couple cases to show that the original requirement was changed and the delivered implementation satisfied the original intent though not the exact working of the requirement, while "Withdrawn" is used when a requirement was removed by Smarter.

As mentioned previously, Fairway did not have access to email communications or meeting notes where decisions might have been made regarding these requirements and therefore can only consider the information provided in the original RFP, proposal and various amendments.

Requirement	Implemented
Item 2: Upload File Types	Yes
Item 3: Single Sign-On (SSO)	Yes
Item 10: Load Test Algorithms	Yes
Item 11: Track Student Items	Yes
Item 12: Calculate Preliminary Scores	Yes
Item 13: Calculate Final Scores	Yes

Item 14: Simulation System UI	Yes
Item 15: Simulation Engine	Yes
Item 16: Simulation Engine	Partial
Item 17: Simulation Engine	Partial
Item 18: Human Scoring Component	Yes
Item 19: Score Objective Test Items	Yes
Item 20: Unscored Items	Yes
Item 21: Transfer Responses	Yes
Item 22: Teacher Hand Scoring	Yes
Item 23: Transfer Responses	Yes
Item 63: Secure delivery of Math and ELA	Yes
Item 64: Add New Tests	Yes
Item 65: Student Response Times	No
Item 66: Review Answers	Yes
Item 67: Check Necessary Components	Yes
Item 68: Secure Delivery to Workstation	Yes
Item 69: Technology-Enhanced Items (TEI)	Partial
Item 70: Student Accommodations	Yes
Item 71: Multimedia and Interactive Items	Yes
Item 72: Automatic Logout	Yes
Item 73: Format Integrity Checks	Partial
Item 74: Annotations and Note Taking	Partial
Item 75: Proctor Print Ability	Yes
Item 76: Review and Change Answers	Yes
Item 77: Support Tools	Yes
Item 78: QTI with APIP Support	Satisfied
Item 79: Manipulatives Support	Yes
Item 80: Practice Test	Yes
Item 81: Practice Test Tutorials	Yes
Item 82: Tools/Tutorials	Yes
Item 83: Foreign Language Support	Yes
Item 84: Non-English Keyboard	Partial

Item 85: Copyrights and Attributions	Yes
Item 86: Capture and Auditing	Partial
Item 87: Psychometric	Partial
Item 88: Alerts	Withdrawn
Item 94: Minimum Specifications	Partial
Item 95: Self-Contained	Yes
Item 96: Multi-tenancy	Partial
Item 97: Coding Standards and Code Reviews	Partial
Item 98: Cloud Hosting	Yes
Item 99: Inter-component Communication	Satisfied
Item 100: Plugin Binary Transport	No
Item 101: Confidentiality of Data	Satisfied
Item 102: Component Security	Yes
Item 103: Secure Access Control	Yes
Item 104: Unique Identifiers	Yes
Item 105: Access Privileges	Yes
Item 106: Federated Identity Management Capable	Yes
Item 108: Require Password Changes	Yes
Item 109: Invalid Login Attempts	Yes
Item 110: Password Entry	Yes
Item 111: Encrypted Password	Yes
Item 112: Secure Login	Yes
Item 113: Unauthorized Access Logging	Yes
Item 114: Security Maintenance Audit Trail	n/a
Item 115: Validation	Satisfied
Item 118: Audits	Partial
Item 119: System Activity Logs	Yes
Item 120: Error Log	Yes
Item 121: Errors Communication	Yes
Item 122: View and Search Error Log	Yes
Amendment 2: English Glossary	Yes
Amendment 3: Test Integration Service	Yes

Amendment 3: Accommodations Changes	Partial
Amendment 3: Embedded Glossary/Thesaurus	Yes
Amendment 6: Teacher Hand-Scoring System	Yes
Amendment 7: Masking Tool	Yes
Amendment 7: Closed Captioning	Yes
Amendment 7: Expanded Item Pool Algorithm	Unknown
Amendment 7: Update Test Shell	Partial
Amendment 7: JAWS	Yes

Table 1: Requirements Summary

Details

In this section, each requirement is described according to the original RFP. The details provided are meant to clarify what has been implemented in relation to the original requirements.

General / All Components

Item 2: Upload File Types

Description System must be able to upload file types as required in the SBAC IT Systems Architecture (Appendix C) for all components.

Implemented Yes

Details The file formats references in the architecture document are as follows:

- Items – APIP
- Item metadata - the defined standard when developed (most likely extensions to IMS QTI / APIP)
- Student Info – SIF
- Student responses – SIF
- Scores – SIF

The items data is implemented using the Smarter Balanced Assessment Item Packaging format while the student data is transmitted between components using the Test Result Transmission format.

Fairway has determined that this change from the original requirements was expected and agreed to, therefore this requirement has been fully implemented using alternative formats.

Portal

Item 3: SSO

Description System must provide a SSO entry point for all SBAC applications. See SBAC IT Systems Architecture (Appendix C).

Implemented Yes

Details Single Sign-On is implemented using OpenAM.

Adaptive Test Engine

Item 10: Load Test Algorithms

Description System must be able to load all test algorithms for each respective test as defined by the CONTRACTORS for RFPs 5, 9 and 20

Implemented Yes

Details The single adaptive algorithm, currently implemented as AIROnline2013, meets the requirements as defined by RFPs 5, 9 and 20, and the test blueprint settings allow the algorithm to be configured on a per test basis.

Currently, the adaptive algorithm is implemented as a collection of a few services. The services implement a generic interface that can be changed by creating a new algorithm in code and using Spring to inject the new algorithm instead of the current one.

It is important to note that while this does allow the algorithm to be changed as needed, only a single instance of an algorithm can be used at a time. The existing algorithm meets the needs as currently defined and would need to be extended to meet future needs instead of loading another complimentary algorithm that could run side by side.

Item 11: Track Student Items

Description System must track student item presentment such that each student can take the same type of test more than once without receiving any duplicate items.

Implemented Yes

Details Fairway reviewed the code for the Adaptive Engine and found that the full history for a testee was retrieved in order to determine which questions had already been asked. The specific logic is located in the AA_GetDataHistory2_SP() method of the ItemSelectionDll package.

Item 12: Calculate Preliminary Scores

Description System must be able to calculate preliminary scores using the raw scores produced from the machine scoring component and associated item score statistics from the Item Authoring and Pool Application.

Implemented Yes

Details The preliminary scores are calculated based on the machine scoring that occurs while the test is taken.

Item 13: Calculate Final Scores

Description System must be able to calculate final scores using ability estimate from machine scored items and results from human scored items and tasks.

Implemented Yes

Details TIS is responsible for taking the machine scored and human scored items and generating the final results.

Item 14: Simulation System UI

Description Simulation system must provide a user interface that allows minimum training for non-technical users and allows for efficient revision of parameters for the purpose of retesting experimental conditions.

Implemented Yes

Details While this requirement is objective, Fairway found that using the application for a few minutes was enough to understand how to create and run a simulation. With minimal training, we feel it would be easy to use for non-technical users.

Item 15: Simulation Engine

Description Provide for ongoing consortium use a simulation engine that executes various algorithms during the development process and simulates events for each test, varying pool and population characteristics as well as CAT constraints.

System must provide an environment for test simulation separate from Test Delivery production environment. The simulation system allows development of new tests and provides a mechanism for evaluating test quality prior to deployment

Implemented Yes

Details The CAT Simulator provides a separate test simulation environment that does not need to be run on the production servers. This simulator allows administrators to

develop new tests, run a specific number of simulated users through the test and evaluate the results through various reports offered.

Test packages can be changed and re-uploaded between test runs, allowing various blueprints to be used when running the simulations. As mentioned in Item 10, there is not a way to easily swap out the entire algorithm without changing the code and redeploying.

Item 16: Simulation Engine

Description Simulation system allows the use of:

- Production, modified or experimental algorithms
- Simulated or actual item pools (when running against actual pools, engine needs to supply information about item load times.)
- Simulated or empirical examinee ability distributions.

Implemented Partial

Details As mentioned in Item 10 above, the use of production, modified or experimental algorithms can be achieved by compiling a new algorithm in code and redeploying the application. It cannot be done by "on the fly" between running various simulations by an administrator using the Simulation application.

Experimental algorithms would require newly developed code to be deployed in order to use the new algorithm since multiple algorithms cannot currently be loaded into the system and used at the same time.

Item 17: Simulation Engine

Description Simulation engine produces output for psychometric and technical analysis including, but not limited to: Simulations of test events for a given ability distribution, item usage, record of information for every simulated response (e.g., current achievement estimate, error of estimate, information range for item search), system response time (e.g. database response time, server response time, and load times for specific items from pilot and field testing)

Implemented Partial

Details After a simulation is run, the following reports are available:

- Summary Stats

- Blueprint Summary
- Field Test Distribution
- Scores
- Opportunities
- Items Report
- Form Distributions
- Test Package

Fairway was not able to find data for the system response times, while the other information is available within the reports listed above.

Human Scoring

Item 18: Human Scoring Component

Description System must able to deliver rubrics to the Human Scoring Component and be able to transfer data received to the Data Warehouse. Please refer to System Architecture Figure 4.3 for an illustration of this data transfer.

Implemented Yes

Details TDS sends the test results to TIS using the Test Results Transmission format. The appropriate items that need to be scored by humans are passed along to the Teacher Hand Scoring System where they are scored by teachers. The teachers are able to view the rubrics for the items in order to score them appropriately.

Machine Scoring

Item 19: Score Objective Test Items

Description System must be able to score objective test items and transfer the data back to the Data Warehouse.

Implemented Yes

Details We have validated that objective test items are scored as the testee answers each question with the scores being stored in the response table of the database and passed along in the TRT file sent to TIS.

Item 20: Unscored Items

Description System must be able to insert unscored items into tests and log them accordingly for research purposes

Implemented Yes

Details The *isfieldtest* flag in the response table is set to true to indicate that the question should not be used when calculating the overall test score.

Distributed Scoring

Item 21: Transfer Responses

Description System must be able to transfer student responses to, and from, the Distributed Scoring component, and relay data back to the Data Warehouse. Please refer to the System Architecture in Figure 4.3 for an illustration of this data transfer.

Implemented Yes

Details Fairway assumes that the Distributed Scoring component is referring to the functionality provided by the Test Integration System. The current flow works as expected. Each finalized test is sent to TIS, which then forwards it to the Data Warehouse when it is completed.

Item 22: Teacher Hand Scoring

Description System must allow teachers to hand score a constructed-response item that may also be scored by Al

Implemented Yes

Details Fairway reviewed the code that processes the TRT file received by the Teacher Hand Scoring System (THSS) and found previously scored answers are ignored during the import process. However, this requirement is specific to the AI scoring algorithm which has intentionally not been implemented at this point. The existing process would work as expected by implementing an AI algorithm that flags the score as tentative instead of being successfully scored. Then it would be available in the teacher queue and available for hand-scoring.

AI Scoring

Item 23: Transfer Responses

Description System must be able to support the Al Scoring component, which will include transferring student responses to the component and relaying data back to that component. Please refer to the System Architecture in Figure 4.3 for an illustration of this data transfer.

Implemented Yes

Details Fairway has confirmed that scoring occurs as the student answers each question and are stored in the response database table as expected.

Test Delivery System

Item 63: Secure Delivery of Math and ELA

Description System is able to securely deliver SBAC English language arts and mathematics tests to students and proctors.

Implemented Yes

Details Fairway has reviewed the architecture, code and communication between systems and has not found any obvious security risks when the system uses HTTPS for all communication. While we believe the system will be deployed this way in practice, the application itself does not enforce access via HTTPS and therefore allows an opportunity to expose a security risk if deployed incorrectly. When configured appropriately the system is able to securely delivery tests to students and proctors.

It is important to note that an in-depth security analysis of the application is outside the scope of this report.

Fairway recommends implementing server-side enforcement that requires all requests be made over HTTPS.

Item 64: Add New Tests

Description System provides ability to add additional tests, specifying subject and grade availability (e.g., science), for secure delivery to students and proctors.

Implemented Yes

Details The system provides the ability to load additional tests as needed.

Item 65: Student Response Times

Description System is able to collect and store individual student response times for each question

Implemented No

Details Fairway was not able to find any tracking of the student response times. We monitored the HTTP traffic sent from the student web browser to the server, reviewed the code and analyzed the response database table. The closest value we could find is the time difference between when the student answered a question and when the question was generated. However, since questions are generated before they are displayed to the student, this value does not represent how long a student takes to respond to a question.

Fairway was able to find the TestShell.Audit javascript code that tracks page loads and durations, however, that data is never sent to the server.

Item 66: Review Answers

Description System allows a student to review their answers for some sections or sets of questions before moving on to the next section or completing the exam.

Implemented Yes

Details A student is allowed to move back and forth between questions within a test segment to review and change their answers as needed before their final submission. This functionality is removed under certain circumstances such as the student pausing the test and waiting too long to return. At the end of each test segment, the student is given another chance to go back to previous questions and review or change answers.

Item 67: Check Necessary Components

Description System is able to check student computers to ensure that they have the components necessary to operate the test.

Implemented Yes

Details In order to ensure that the student can hear audio files that are available during the test, the application prompts the user with an audio file and confirms they can hear it before they are allowed to move on. Similarly, if the text-to-speech accommodation is assigned to a student, the system validates that a secure browser is being used and prompts the student to confirm that they can hear the text-to-speech example being read.

Item 68: Secure Delivery to Workstation

Description System has the ability to deliver the test software securely to individual student workstations / devices with minimal total cost of ownership and stability that is buffered against software updates.

Implemented Yes

Details As mentioned in Item 63, the test vendor may deploy non-secure connections and the application does not prohibit those decisions.

The total cost of ownership and system stability are areas that are subjective and difficult for Fairway to evaluate appropriately and are therefore not addressed specifically in reviewing this item.

Item 69: Technology-Enhanced Items (TEI)

Description System will be able to deliver TEIs as defined in Proposed TEI Specifications (see Appendix B)

Implemented Partial

Details Fairway reviewed each item template listed in the original proposal and found example implementations of most types or determined they would be possible based on the graphic response type. AIR recommended a general shape tool instead of individual circle or parabola tools. We were not able to find a general shape tool or specific implementations for circle or parabola. Additionally, we determined that the simulation type was not implemented.

During the life of the project, Smarter withdrew the simulation type requirement and choose not to use the circle and parabola tools.

See Table 2 for a more detailed breakdown.

Item Template Type	Implemented
Select Defined Partitions (Hot Spot)	Yes
Select Objects (Hot Spot)	Yes
Selecting Points and Ranges on Number Lines (Hot Spot)	Yes
Classification (Drag and Drop)	Yes
Select and Order (Drag and Drop)	Yes
Connections (Drag and Drop)	Yes
Tiling (Drag and Drop)	Yes
Object Transform (Graphic Response)	Yes
Partition Number Line Then Place Points (Graphic Response)	Yes
Single Angle (Graphic Response)	Yes
Single Ray (Graphic Response)	Yes
Single Circle (Graphic Response)	Unused
Single Parabola (Graphic Response)	Unused
Partition Object Then Select (Graphic Response)	Yes
Multiple Lines Then Select (Graphic Response)	Yes
Place Points (Graphic Response)	Yes
Single Line (Graphic Response)	Yes
Single Line Then Select (Graphic Response)	Yes
Vertex-Based Quadrilaterals (Graphic Response)	Yes
Vertex-Based Triangles (Graphic Response)	Yes
Reorder Text (Hot Text Manipulation)	Yes
Select Text (Hot Text Manipulation)	Yes
Drop Downs (Hot Text Manipulation)	Yes
Equation Editor	Yes
Simulation	Withdrawn

Table 2: Technology-Enhanced Items

Item 70: Student Accommodations

Description System supports items/tasks that can accommodate students with disabilities (e.g. items/tasks including Sign Language, refreshable Braille, text-to-speech tags, text magnifying software, and speech-to-text tags). Braille support must include contracted, uncontracted and Nemeth Braille.

Implemented Yes

Details The specific accommodations and Braille support listed have been implemented as expected. The accommodations are assigned to a student within ART and are pulled into TDS when the student logs in.

Item 71: Multimedia and Interactive Items

Description System supports multimedia and interactive items that are written in HTML 4 with HTML 5 extensions.

Implemented Yes

Details The application supports multimedia and interactive items as expected.

Item 72: Automatic Logout

Description System automatically logs out after a period of no activity. For certain users (e.g. students with accommodations needs), system can restart after a period of no activity, and resume the session where it left off. Activity timing must be variable based on student profile and activity required must be minimal so as not to interfere with students' testing

Implemented Yes

Details Fairway has observed these behaviors in our development environment and have confirmed the process works as expected.

Item 73: Format Integrity Checks

Description System has the ability to upload and conduct format integrity checks on files as necessary to support test registration, interim assessment scoring and possible extensions of the summative assessment.

Implemented Partial

Details The Test Results Transmission (TRT) XML file is sent from TDS to the Test Integration System (TIS) and from TIS to the Teacher Hand Scoring System (THSS). TIS does not perform validation against the XSD schema and instead checks that the root element is TDSReport. There is a TODO comment in the code mentioning adding validation using the XSD schema.

When the TRT file is sent to THSS, it is validated against the XSD to make sure it is an appropriate file type and reports back any validation errors that are found.

Item 74: Annotations and Note Taking

Description System supports electronic annotation of items/tasks, and note-taking, as a part of the item response. Annotations and note-taking must persist and be available when a student reviews the item. For the practice test, the persistence would last only for the duration of the current session.

Implemented Partial

Details TDS implements a note-taking feature that is controlled by a specific universal accommodation. When enabled, a notepad option is available and allows the student to enter and save notes that are associated with a specific item of the test. The notes are saved while you are taking the test and are shown when you go back to the question while still within a test segment.

However, when pausing a test and coming back a minute later, the notes are not available anymore. The requirement suggests to us that the note should persist even when taking a break and the "Usability, Accessibility, and Accommodations Guidelines" from August 25, 2015 (http://www.smarterbalanced.org/wordpress/wp-content/uploads/2014/08/SmarterBalanced_Guidelines.pdf) confirm that. From the document, the Digital Notepad is described as follows: "This tool is used for making notes about an item. The digital notepad is item-specific and is available through the end of the test segment. Notes are not saved when the student moves on to the next segment or after a break of more than 20 minutes."

Annotations supported by TDS include highlighting, strikethrough, and mark for review. Like notes, these annotations are available within the same session but are not persisted when a student takes a break of any length. It is important to note that the Global Notes feature does persist between sessions unlike the individual item notes feature.

Item 75: Proctor Print Ability

Description System allows test administrators to print items, stimuli, and necessary resources to support system accessibility requirements, with appropriate security procedures.

Implemented Yes

Details The application has implemented the ability for test administrators to print resources as needed in a secure manner.

Item 76: Review and Change Answers

Description System allows a student to review and change their answers according to specifications identified in RFPs: 5, 9, and 20.

Implemented Yes

Details TDS does support students reviewing and changing their answers within a test segment. Please refer to Item 66 above for more details.

Item 77: Support Tools

Description System support tools being tagged to specific items.

Implemented Yes

Details The proposal submitted by AIR recommended against having support tools tied to a specific item and advised having the supports tied to the test or test segment. AIR argued that it would be less confusing to the user to have the options be shown and then hidden while going from item to item.

Fairway assumes that this approach was agreed upon since that is what is currently implemented within the application. Therefore, we agree that the support tool requirement was satisfied.

Item 78: QTI with APIP Support

Description System supports QTI with APIP extensions protocols.

Implemented Satisfied

Details TDS does not actually support QTI with APIP because it was determined that it did not adequately support the types of assessment items needed. AIR and Smarter then agreed to develop a new format called the Smarter Balanced Assessment Item Packaging Format and work with IMS Global to develop a Smarter Balanced Profile for IMS APIP.

Item 79: Manipulatives Support

Description System supports manipulatives (e.g., calculator, spell check, graphing tools, visually based dictionary, dictionary, thesaurus, text pop out, measurement tools, electronic annotation, formula charts, and sketch pads) being tagged to specific items/tasks during the test.

Implemented Yes

Details The Consortium and AIR agreed to the features that should be available in the system. In addition, the Consortium established the list of universal tools that students may use during the test. The requirements for universal tools and designated supports are defined in the "Usability, Accessibility, and Accommodations Guidelines" document (http://www.smarterbalanced.org/wp-content/uploads/2015/09/Usability-Accessibility-Accomodations-Guidelines.pdf). Each of the requirements in this document has been implemented as expected.

The Embedded Universal Tools are shown in Table 3 below:

Embedded Universal Tool	Implemented
Breaks	Yes
Calculator	Yes
Digital Notepad	Yes
English Dictionary	Yes
English Glossary	Yes
Expandable Passages	Yes
Global Notes	Yes
Highlighter	Yes
Keyboard Navigation	Yes
Mark for Review	Yes
Math Tools	Yes
Spell Check	Yes
Strikethrough	Yes
Writing Tools	Yes
Zoom	Yes

Table 3: Embedded Universal Tools

The Embedded Designated Supports are list in Table 4 below:

Embedded Designated Support	Implemented
Color Contrast	Yes
Masking	Yes
Text-to-Speech	Yes
Translated Test Directions	Yes
Translations (glossaries)	Yes
Translations (stacked)	Yes
Turn Off any Universal Tool	Yes

Table 4: Embedded Designated Supports

The Embedded Accommodations are shown in Table 5 below:

Embedded Accommodation	Implemented
American Sign Language (ASL)	Yes
Braille	Yes
Closed Captioning	Yes
Streamline	Yes
Text-to-Speech	Yes

Table 5: Embedded Accommodations

Item 80: Practice Test

Description System is able to administer a practice test with all the functionality of a real test.

Implemented Yes

Details The practice tests are capable of all the functionality of a real test except for one specific area. When a Guest user is accessing the practice test, they are not able to pause the test and rejoin this same test later. Since they are not a specific user, it is not possible to track the user between sessions to know they should join and restart a specific test opportunity. We feel this is to be expected and therefore the requirement is satisfied.

Item 81: Practice Test Tutorials

Description System supports tutorials being tagged to items/tasks, which the student can access during the practice test.

Implemented Yes

Details While taking the practice test, a user is presented with a Tutorial option. When selected, a dialog is presented with information on the specific type of item that is currently being displayed.

Item 82: Tools / Tutorials

Description System is able to provide tools / tutorials to help students during the test event (e.g. TEI item manipulations) within the testing environment

Implemented Yes

Details The application provides tools and tutorials to individual students based on accommodations assigned in ART. This is working as expected.

Item 83: Foreign Language Support

Description System is able to deliver items/tasks that have been translated into other foreign languages including languages that use ASCII- and non-ASCII-based characters.

Implemented Yes

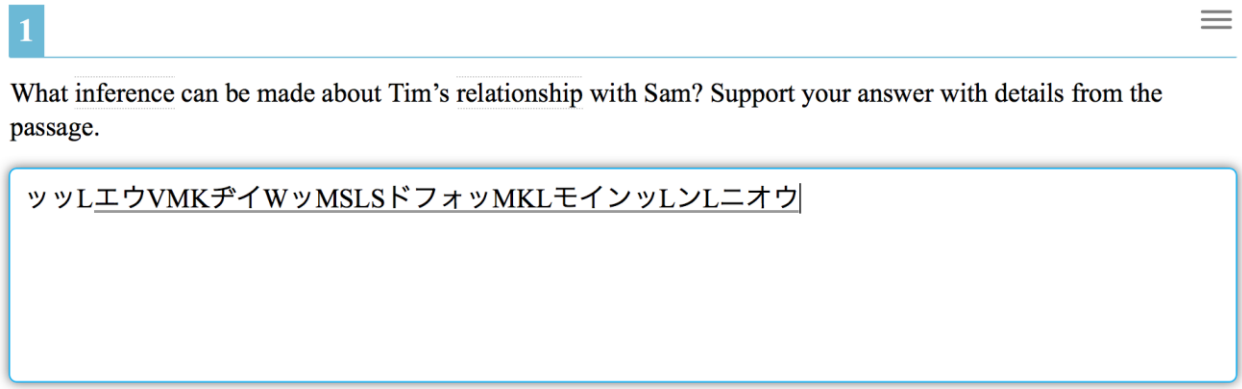
Details We have confirmed that it is possible to display the test in Spanish using the Language accommodation options available and we see no reason why the test could not be delivered in other languages as needed.

Item 84: Non-English Keyboard

Description System is able to function with non-English keyboards.

Implemented Partial

Details Fairway used a virtual keyboard to answer questions in both Spanish and Japanese. The application is able to accept Spanish characters and handle them correctly. However, when answering questions in Japanese the data is not stored correctly in the database and when refreshing the page, therefore retrieving the answer from



the server, the characters are not displayed correctly. The following screenshots show the data that was entered and then the data when it was redisplayed.

Item 85: Copyrights and Attributions

Figure 1: Japanese Answer Before Saving

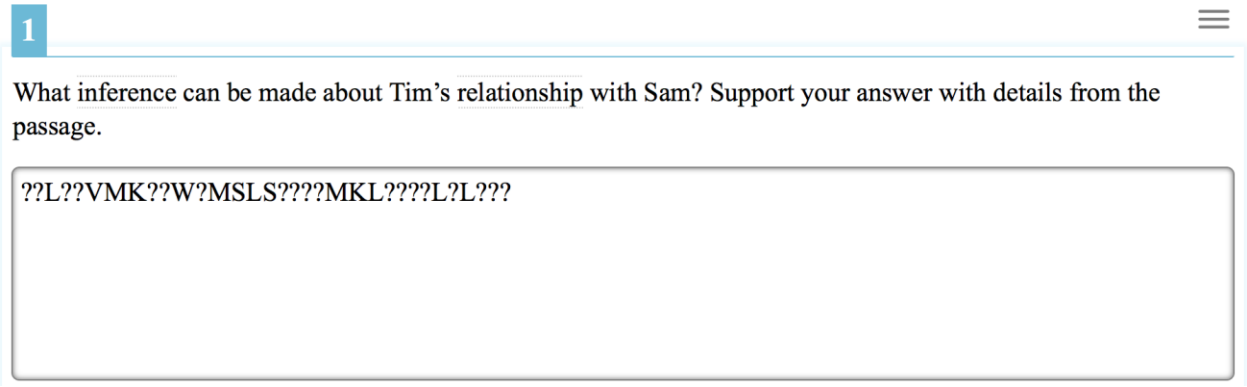


Figure 2: Japanese Answer Retrieved from Database

Description System is capable of displaying copyrights and attributions

Implemented Yes

Details While Fairway has not specifically seen copyrights or attributions while using the practice/training and IRP tests, we see no reason this cannot be easily implemented. The content of each item is stored in the test package XML file as HTML within a CDATA block. Therefore, the copyright and attribution information could be stored in the HTML and be displayed during the test as expected.

Response Data Store

Item 86: Capture and Auditing

Description System captures student's response data and is available for auditing (e.g., key stroke, deleting, and response deletion).

Implemented Partial

Details The system captures the student's response data and keeps an audit trail for each change the student makes to responses during the test. The audit information is stored in the *session.testeeresponseaudit* table. We are unable to find any data that would allow an administrator to track keystroke level data.

Item 87: Psychometric

Description System is able to capture and deliver to the Response Data store for psychometric use as specified in RFPs 5, 9, and 20.

Implemented Partial

Details Page 112 of the original AIR proposal states, "Store other information about how the student responded (e.g., time to answer, time to render). All student telemetry data, system telemetry data, and system logging information will be centrally stored on the backend of the system. For security and reliability reasons, these data will be processed in a different fashion than response data but will still be complete and up-to-date. Along with psychometric analysis purposes, these data can be used to "replay" the student experience."

We have analyzed the HTTP traffic from the client's browser to the server and have not been able to find any psychometric data being sent that would allow the student activity to be replayed. As mentioned previously, we do not see a way to determine the student's time to answer. We analyzed the database tables to see if there is a logical storage location for psychometric data and were not able to find that either.

The database is storing the history of student responses. Therefore, if a student returns to a question and changes it, it is tracked in the database. These data allow for the ability to determine if a student has changed answers and the order in which they provided answers.

Item 88: Alerts

Description System includes a suite of alerts to the test administrator and Consortium delegates if there appears to be a testing irregularity

Implemented Withdrawn

Details We have not seen any evidence that the system checks for testing irregularities and alerts the test administrator.

Smarter Balanced indicates that this requirement was withdrawn.

Systems Architecture

Item 94: Minimum Specifications

Description The proposed system shall operate efficiently with the SBAC's minimum software and hardware specifications. SBAC's minimum specs will be posted on SBAC's website in March.

Implemented Partial

Details The "Smarter Balanced Technology Strategy Framework and Testing Device Requirements" document dated November 1, 2013 defines the minimum requirements for running the secure browser in order to take a test. Fairway installed the secure browser on Windows XP, Mac OS X and Ubuntu 10, and took a practice test in order to validate everything worked with the minimum specifications. In order to match the RAM and CPU requirements as closely as possible, we used VirtualBox. The RAM was set to the exact minimum requirement while the CPU was throttled by giving the virtual machine an execution cap for the processor. The results are shown in Table 6 below.

Operating System	Minimum Requirements	Result
Windows XP	233 MHz processor	Worked as expected

(service pack 3)	128 MB RAM	
Mac OS X 10.4	300 MHz processor 256 MB RAM	Fairway only had access to version 10.6 of Mac OS X which has a minimum 1 GB RAM requirement. Therefore, we were unable to test version 10.4 and 256 MB RAM. The system worked as expected with 1 GB of RAM and a 300 MHz processor on Mac OS X 10.6.
Ubuntu 10.04 LTS	233 MHz processor 64 MB RAM	Unable to install Ubuntu with 64 MB RAM in VirtualBox. Worked as expected with 128 MB RAM.

Table 6: Minimum Requirements

Item 95: Self-Contained

Description System is fully self-contained and capable of being operated by SBAC with no dependency on vendor services for its routine operation.

Implemented Yes

Details The system is capable of operating without a dependence on the vendor for services.

Programming Standards

Item 96: Multi-tenancy

Description System supports multi-tenancy.

Implemented Partial

Details While testing the multi-tenancy support in TDS, Fairway found limitations in the implementation that don't allow for true multi-tenant support. We believe that true multi-tenant support would allow for multiple independent states to be run from a single TDS installation. While configuring our environment to be used by two states we ran into the following issues:

- Permissions does not allow for multiple tenants, meaning all tenants must use the same permission rules
- TestSpecBank "side loading" process does not allow for two tenants to use the same test
- Student and Proctor applications are configured in ProgMan for a specific state code, so a proctor of a different state will never see tests when logging in

We were able to find some work arounds during our attempt to configure two different state tenants, and loaded a test into ART with eligible students. However, multiple Student and Proctor servers would need to be deployed in order to work for multiple states, though they could conceivably use the same shared services.

Item 97: Coding Standards and Code Reviews

Description The Vendor uses coding standards and code reviews to ensure that the development team follows these coding guidelines to facilitate the readability of the source code and to make software maintenance easier.

Implemented Partial

Details AIR's original proposal outlines the responsibilities and importance of performing code reviews during the development process, and there is no reason to believe that this did not occur. We have no way of knowing AIR's internal coding standards and whether they were followed during development.

While code readability and ease of software maintenance are subjective, Fairway feels that the organization and inter-dependencies of projects makes the application difficult to maintain. As an example, there are 18 repositories that are needed in order to build the components required to run TDS. The shared artifacts are setup in a manner that makes it difficult to manage each service independent of the others. For example, it is not possible to upgrade Spring in the Student application without upgrading Spring in the Proctor application. This does not match the intent of the "Smarter Balanced System Architecture and Technology Report" which defines independent components that are self-contained and communicate with other components through well-defined protocols.

Additionally, the Student and Proctor applications appear to be a direct class by class ports from AIR's proprietary C# application to Java. This led to many things being done the "C# way" instead of the "Java way." Many classes that do not exist

in Java were recreated, such as HttpContext and SqlConnection, presumably to make the port from C# to Java simpler, instead of rewriting areas to use Java best practices.

Many naming conventions and file extensions from .NET were left in place causing some confusion for those new to the codebase. The API endpoints still have the AXD extension (used for .NET Handlers) and the login and test shell pages use the ASPX extension (used for .NET WebForms pages) when the underlying file is XHTML used by Java Server Faces. Additionally, code used during the porting process has been left in place though it is unused. This has caused confusion while reviewing the code to understand and get familiar with the application.

The items listed here contribute to creating a steep learning curve in order to work and maintain the TDS codebase.

Item 98: Cloud Hosting

Description System is able to support cloud-based hosting services

Implemented Yes

Details Fairway has successfully deployed the TDS application in the AWS cloud utilizing Elastic Load Balancing and MySQL RDS. Having worked in the code and seen the architecture of the application, there is no reason why TDS would not work well in a cloud hosting environment.

Item 99: Inter-component Communication

Description System's inter-component communication uses current standards (e.g. SIF, IMS, etc.) that support the SBAC IT System Architecture (see Appendix C).

Implemented Satisfied

Details As mentioned previously, it appears that the formats used were changed from standards such as SIF and QTI with APIP, to using new Smarter Balanced created formats. Since this approach was agreed upon between Smarter and AIR, this requirement is implemented as expected.

Item 100: Plugin Binary Transport

Description System supports the Plugin Binary Transport, which requires an abstract API to be developed that components can call with a consistent interface. The API uses the data format described by the accepted standard for that assets domain (i.e. item format, student information, and student response, etc.).

Implemented No

Details The Smarter Balanced Architecture Report (http://www.smarterbalanced.org/wordpress/wp-content/uploads/2012/03/SmarterBalanced_ArchitectureReport_120321.pdf) describes a pluggable interface on page 25 that allows the transport protocol to be changed via a configuration file. The current application defines the protocols in advance and hard codes the SFTP or REST access. The endpoints are configurable by changing the URLs in Program Management (ProgMan), but the protocols used are defined in advanced and cannot be changed without changing the code.

Security/Access Control

Item 101: Confidentiality of Data

Description System ensures that the integrity and confidentiality of data is protected by safeguards to prevent release of information without proper consent.

Implemented Satisfied

Details Much of the responsibility in safeguarding confidential data lays with the security of the servers themselves. The system administrator must implement best practices in locking down the servers; only opening ports and access where absolutely necessary, utilizing the latest standards for SSL encryption and hashing and implementing strong passwords.

Before development began, the Smarter Balanced Architecture Working Group approved the recommendation to not encrypt data at rest and instead rely on the physical security of the data centers and/or file system encryption.

It is important to determine what data are confidential and the sensitivity level of that data. For example, if a social security or credit card number is being stored at rest in a database it is considered best practice to encrypt that information. If the database is compromised, having encrypted data adds another level of protection.

Fairway assumes that the student data and the test responses are the types of data that would be considered most sensitive. In the TDS database the student name, date of birth, gender, race, migrant status, economic status, disabilities and the student identifier (potential social security number if the school uses that) are all stored at rest and left in plain text. The test responses are also stored unencrypted but are assumed to be less sensitive than the student demographic information.

Having this information stored in plain text is a potential security concern and an area where an additional safeguard could be added to help prevent the release of information without the proper consent if the information is deemed sensitive by Smarter Balanced.

Item 102: Component Security

Description System provides security for all system components (see Figure 4.3) at database, workstation, and individual operator levels.

Implemented Yes

Details As mentioned previously, much of the responsibility for securing the application, workstation and database is with the server and network security setup when deploying the application. Assuming that the system administrator implements bests practices in locking down the servers, Fairway does not see any obvious security risks with the architecture and design of the TDS application. However, an in-depth security analysis is outside the scope of this report.

Item 103: Secure Access Control

Description System provides secure access control based upon single unique user login.

Implemented Yes

Details TDS utilizes SAML via OpenAM to provide secure access to each component based on a single user login. Role based permissions are controlled in the Permissions component and are used to determine the type of access each user is given.

Item 104: Unique Identifiers

Description System provides secure unique identifiers for all users.

Implemented Yes

Details OpenDJ generates a random UUID value which is stored in a property called *sbacUUID*. This value is then used when accessing user information within TDS and is the link between the TDS database and the OpenDJ credentials store.

It is important to note that the term "users" refers to proctors and administrators and not students. The students are assigned an SSID within ART and a unique incremental integer from the database.

Item 105: Access Privileges

Description System checks each user's access privileges at login, and automatically disables or enables client functions (in real time) based upon the user's profile.

Implemented Yes

Details After a user logs in, the tenancy chain, containing each role the user belongs to, is supplied to the specific TDS component being accessed. The roles are then used to determine the specific access in conjunction with the rules configured in the Permissions component. This is done in real time when the user accesses each component.

Item 106: Federated Identity Management

Description System provides federated identity management capability.

Implemented Yes

Details The application is setup using SAML through OpenAM. This allows for the implementation of federated identity management across multiple organizations.

Security/Password Controls

Item 108: Require Password Changes

Description System provides an options to require user passwords to be automatically prompted for change after a configurable, defined period has passed, such as 30, 60, or 90 days.

Implemented Yes

Details OpenDJ offers the ability to require a user to change their password after a predetermined amount of time.

Item 109: Invalid Login Attempts

Description System provides the option to disable user IDs for a configurable time frame after a specified number (configurable) of consecutive invalid login attempts.

Implemented Yes

Details OpenAM can be configured to lockout a user after a certain number of failed login attempts. In the version installed during TDS deployment, this feature is not enabled by default, but can be easily turned on as needed.

Item 110: Password Entry

Description System enters passwords in a non-display field.

Implemented Yes

Details Password fields are configured to hide the display of the password as it is entered.

Item 111: Encrypted Password

Description System encrypts passwords when they are routed over the network.

Implemented Yes

Details Assuming proper deployment and configuration, the login screen is accessed via HTTPS, and therefore, passwords are encrypted as they are sent from the client's browser to the server. Additionally, passwords are not stored in the cookie directly or sent between servers when doing REST API calls.

Item 112: Secure Login

Description System provides a method of secure login for students and adults and comprehensive security for all system components.

Implemented Yes

Details Much of the system security depends on the security and configuration of the servers where the application is deployed. Unfortunately, the application itself cannot control for this. We previously provided a recommendation for strictly enforcing access via HTTPS by the application itself in order to mandate that secure access is provided.

Fairway has not found any obvious security issues in the application design or code regarding the login or overall system security assuming all client-server and server-server communications are conducted using SSL.

Security/Activity Logging

Item 113: Unauthorized Access Logging

Description System logs unauthorized access attempts by date, time, user ID, device, and location.

Implemented Yes

Details Invalid login attempts are logged within OpenAM in the amAuthentication.error log file.

Item 114: Security Maintenance Audit Trail

Description System maintains an audit trail of all security maintenance performed by date, time, user ID, device, and location, with easy access to information.

Implemented n/a

Details The application itself does not provide a mechanism for security maintenance or upgrades and the responsibility lies on the system administrator to manage that process. Therefore, there is not an audit trail that keeps track of this within the applications themselves.

Edit and Validation Control

Item 115: Validation

Description System includes comprehensive field edits to prevent incomplete or incorrect data from entering the system.

Implemented Satisfied

Details The Student application does provide some client side validation meant to prevent incorrect data entry. The student is required to answer each question on the page before they can move next, and the application prompts the student when they enter a character like “a” into the field when it expects a number.

However, there are gaps that allow the invalid data to be submitted to the server. Even though the student is told that they must enter numerical data and the field turns red, after clicking OK, the student is allowed to move to the next page and submit the test at the end. The APIs themselves do not validate the data and relies on the client side validation.

It is important to note that the system was calibrated using AIR’s proprietary system during the Field Test and the requirement to maintain the same validation and user experience in the open source system superseded this specific requirement. Therefore, the implementation matches Smarter Balanced expectations.

System Auditing

Item 118: Audits

Description Sufficient audits must be available to identify the source and time of data changes related to system components.

Implemented Partial

Details To make data changes affecting the system components (outside of admin tasks on the servers themselves) a user will either alter the configuration data via the Program Management (ProgMan) component or make changes to the configuration tables directly in the database. Any changes made within ProgMan are saved into an audit collection in MongoDB called *propertyConfigAudit* and can be easily monitored or audited.

However, only being able to change much of the configuration data by accessing the database directly does not allow for a proper audit table to be implemented. Therefore, changes made to configuration tables are not currently tracked and unable to be audited.

Item 119: System Activity Logs

Description System must ensure that it logs system activity necessary to monitor and debug the system in a timely and accurate manner.

Implemented Yes

Details Each component within TDS logs detailed information to the system log files on the server, including each SQL statement performed. While it is certainly subjective, sometimes it is necessary to have audit tables track the change of data over time in order to debug certain issues in a timely manner. To this end, the Student application stores audit information for test sessions and test opportunities in order to track the state changes of each item. The various states are listed in Table 7 below.

The Permissions component does have the system log files, however there don't appear to be any audit tables tracking the changes made to user permissions.

Table	Possible States Tracked
archive.opportunityaudit	approved, completed, denied, paused, paused by session, pending, restart [num], review, started, suspended
archive.sessionaudit	administratively closed, closed, Handoff, open, TACheckin TIMEOUT

Table 7: TDS Audit Tables

Error Handling

Item 120: Error Log

Description System must ensure that all errors are written to an error log.

Implemented Yes

Details All errors are written to log files on each server following a common approach to logging for java web applications.

Item 121: Errors Communication

Description Errors to the end user must be communicated in plain language with an explanation of required action.

Implemented Yes

Details Fairway reviewed the error message templates and they appeared to be well constructed and easy to understand. Additionally, each error message we received when using the application met this requirement as well.

Item 122: View and Search Error Log

Description System must allow for a system administrator to view, filter, sort, and search the error log.

Implemented Yes

Details In addition to being written to an error log file on disk, each system error is logged in the archive.systemerrors table. This allows an easier way for administrators to view, query and sort the errors.

Amendments

Amendment 1

Pilot Test Concurrent Users

Description The pilot test requirements were increased from 10,000 concurrent users to 40,000 concurrent users.

Implemented n/a

Details This requirement is not relevant to the TDS application that was delivered since the pilot test utilized a combination of proprietary and newly developed software.

Amendment 2

English Glossary

Description Provide English glossary access tools for two forms (grade 3 math and grade 4 English language arts) as part of the test engine delivery application.

Implemented Yes

Details TDS offers English glossaries for certain terms defined in the item metadata. The specific term is highlighted and a dialog is displayed with a definition when the student clicks on the word. The grade 3 math and grade 4 English language arts practice tests utilize this feature, though it is not limited to those specific tests. Any test can be configured to make English glossaries available to the students.

Amendment 3

User Preferences and Identifier Management

Description Removed User Preferences and Identifier Management from the list of components that would be developed. These components were originally listed in the "Portal and Shared Services" category.

Implemented n/a

Details These components were not developed.

Test Integration Service

Description The requirement to develop the Test Scoring and Test Integrator components is defined in Amendment 3 and the workflow from test delivery, to test scoring and finally to the data warehouse is updated to include TIS.

The Test Scoring Component is responsible for taking all item scores of a student's test opportunity and then scoring the test. This includes scores for any reporting categories; including the strands, standards and/or benchmarks.

The Test Integrator is responsible for the "automated virtual stitching" of the test where there are hand-scored responses. Since hand-scored items go through a different process than machine-scored items, TIC takes the machine-scored items of a test opportunity and integrates them with the hand-scored items of the same test in an XML file. This file will then be uploaded into the Data Warehouse.

Implemented Yes

Details The Test Integration Service is responsible for the test scoring and "virtual stitching" of test where there are hand-scored items by integrating with THSS.

TIS includes multiple components that handle the various responsibilities of the overall service. The TDS Reciever ingests test results from TDS storing them in the database to be handled by the TIS Service. The TIS Service monitors the database, sending tests to be hand-scored when needed and triggering the final test scoring

when all of the individual responses have been scored. The Scoring Daemon then computes the final test score and sends the final test result to the Data Warehouse.

Accommodations Changes

Description Non-embedded accommodations to be added to the accommodations upload spreadsheet to allow system to capture the user of both embedded and non-embedded features.

Enable users to turn-off English glossary for the student in the ART (originally called TIDE) UI as well as through the accommodations upload spreadsheet.

Implemented Partial

Details Non-embedded accommodations are available to be uploaded via the configuration file and can be tracked on a per student basis. ART provides non-embedded options that can be selected when editing a student. There is also an "Other" field where free text can be entered, however, we did find a bug using this. The text is captured and available in ART, but the "Other" information is not shown to the proctor when viewing the student details. It always displays "None" next to the "Other" label.

The student upload file does not appear to be working as expected. When accommodations are included for the students, the system displays the error "The header is not applicable for his format." In order to upload students, we had to remove all accommodation fields.

Embedded Dictionary/Thesaurus

Description Support Embedded Dictionary/Thesaurus for 6m students

Implemented Yes

Details Fairway has seen the embedded dictionary and thesaurus in the practice and training tests. AIR brokered a license to use the Merriam Webster online dictionary and thesaurus which is capable of supporting more than 6 million students.

Amendment 4

Digital Library Provisioning Process

Description The following list are the Digital Library provisioning process requirements as defined in Amendment 4:

- Supporting only the association of users to state
- Modifying SSO to accept users directly (csv, tab delimited) rather than XML
- Launching Core Standards and SSO for Digital Library
- User Manual
- Support through the end of the contract

Implemented n/a

Details The Digital Library is outside the scope of this report.

Amendment 5

Additional Field Test Events

Description Provide services as specified earlier in this contract related to the Smarter Balanced Field Test to support the inclusion of additional student testing events, irrespective of content area.

Implemented n/a

Details The Field Test testing events are outside the scope of this report.

Amendment 6

Teacher Hand-Scoring System

Description Develop Interim Local Scoring System: a scalable platform for teachers to hand score interim assessment items.

Implemented Yes

Details The following table defines the specific features that were required when developing the Teacher Hand-Scoring System (THSS).

Requirement	Implemented
Automatically transfer item responses and scores	Yes
View summary of scoring queue	Yes
Reassign items to peer scorers	Yes
View responses, exemplars, etc. and score items	Yes
Administrative scorer console for managing scoring assignments	Yes

Teacher dashboard page	Yes
Admin dashboard page	Yes
Scoring page	Yes
Peer reassignment of items to other scorers	Yes
Distribution system: scalable system for distributing scoring tasks among multiple servers	Yes
Distribution database: store scored and un-scored results	Yes
Scoring servers: scaled servers to support teacher scoring	Yes
Queue: decouple Test Delivery from Test Integration	Yes
SFTP: transfer mechanism to Data Warehouse	Yes

Amendment 7

Masking Tool

Description Creation, Enhancements and Fixes to Masking Tool

Implemented Yes

Details Fairway was not provided additional details for this requirement, therefore we are evaluating the requirement as the development and delivery of the Masking Tool. The Masking Tool is available when the appropriate accommodations are granted and performs as expected, blocking out parts of the page as drawn by the student.

Closed Captioning

Description Creation, Enhancements and Closed Captioning

Implemented Yes

Details Fairway was not provided additional details for this requirement, therefore we are evaluating the requirement as the development and delivery of the Closed Captioning feature. We have used the Closed Captioning feature while taking practice tests and it works as expected.

Deploy Practice/Training Test

Description Deploy and maintain Practice/Training Test on Open Source System through 12/31/2014

Implemented n/a

Details This is outside the scope of this report.

Bug Fixes and Support

Description Provide ongoing bug fixes, enhancements, and ongoing support to Smarter Balanced – contract close-out, support transition to open source

Implemented n/a

Details This is outside the scope of this report.

Test Packaging

Description

- Test Packaging Summative for 2014/2015 Operational Administration
- Test Packaging Interim Summative Clone Fixed
- Test Packaging Interim Blocks Fixed
- Raw Item Content Test Package

Implemented n/a

Details This is outside the scope of this report.

Expanded Item Pool Algorithm

Description Expanded Item Pool Algorithm Update

Implemented Unknown

Details Fairway was not able to find more specific details for this requirement and therefore is unable to determine if it was implemented as expected.

Translated Rubrics

Description Testing Translated Rubrics

Implemented n/a

Details This is outside the scope of this report.

Update Test Shell

Description Update the test shell to the “universal” shell to make it WCAG 2.0 compliant

Implemented Partial

Details Fairway reviewed TestShell.xhtml and it appears to show that it was upgraded to the "universal" shell. The assets live in a directory named "Universal" and the title of the page is "Universal Shell."

We are unsure if the expectation of the requirement was to achieve WCAG 2.0 compliance or to change the test shell in the hopes that the "universal" shell would be compliant. We feel the intent of the requirement was to achieve compliance. Speaking with Smarter Balanced we have learned that certification was never attempted.

We found many accessibility issues when using the application outside of "streamlined mode," and assume that "streamlined mode" is required in order to attempt WCAG 2.0 compliance. However, even in "streamlined mode" we have found an instance where accessibility features are lacking. When using the keyboard to access the "hamburger" options menu, using the enter and spacebar keys to select a menu option do not work and therefore the options cannot be selected without using the mouse.

JAWS

Description Support additional work to enable JAWS to work for 2014-15

Implemented Yes

Details The student application works as expected with JAWS and the HTML uses the aria attributes as expected.

Content Cleanup

Description Support final content cleanup of items to meet expected specifications

Implemented n/a

Details This is outside the scope of this report.

References

Source	Description
SBAC RFP No. 11	The original RFP released by Smarter Balanced
SBAC RFP No. 11 Proposal by AIR	AIR's proposal
SBAC RFP No. 11 - Amendment 1*	Amendment executed on November 6, 21012
SBAC RFP No. 11 - Amendment 2*	Amendment executed on January 4, 2013
SBAC RFP No. 11 - Amendment 3*	Amendment executed on March 7, 2014
SBAC RFP No. 11 - Amendment 4*	Amendment executed on July 22, 2014
SBAC RFP No. 11 - Amendment 5*	Amendment executed on July 17, 2014
SBAC RFP No. 11 - Amendment 6*	Amendment executed on September 30, 2014
SBAC RFP No. 11 - Amendment 7*	Amendment executed on December 23, 2014
Usability, Accessibility, and Accommodations Guidelines	These guidelines document the universal tools, supports and accommodations of the TDS application
Student Application Source Code	student_release BitBucket repository
TDS / Proctor Source Code	tds_release BitBucket repository
Database Access Source Code	tdsdll_release BitBucket repository
TIS Source Code	testintegrationsystem_release BitBucket repository
Teacher Hand Scoring System Source Code	TeacherHandScoreSys_release BitBucket repository
Student Report Processor Source Code	StudentReportProcessor_release BitBucket repository
CAT Simulator Source Code	CATsimulator_release BitBucket repository
Item Renderer Source Code	ItemRenderer_release BitBucket repository
Item Selection Source Code	itemselectionshell_release BitBucket repository
OpenDJ Source Code	opendj_release BitBucket repository
OpenAM Source Code	openam12_release BitBucket repository

Table 8: Source Documents

* The amendments were provided to Fairway by Smarter Balanced.