Test plan

Smarter Balanced Assessment Consortium Test Delivery System

Components: Test Authoring, Test Item Bank, Test Spec Bank

Smarter Balanced Task Orders 02 and 03

Revision History

Revision Description	Author	Reviewer	Date
0.1 – First Draft	Peter Tereshchenko		Feb. 10, 2016
0.2 – Second Draft	Peter Tereshchenko	Rami Levy	Feb. 26, 2016
1.0 – Final version	Peter Tereshchenko	Rami Levy	Feb. 29, 2016

Contents

Tables

1 Introduction and Background of Systems

The purpose of this document is to define testing scope and approach for Task Orders 02 and 03. These task orders affect only certain functions of CS, TIB, TA, TSB and TDS. This test plan is focused on testing only the affected parts of the functionality of above components and is not intended to cover, either in part or in full, other functionality.

Component	Abbreviation	Description
Test Authoring	*TA*	*Test Authoring is a browser-based interface that allows users to create tests, segments, blueprints and other elements that comprise a test. It provides a means of validating tests and an approval workflow.*
Test Packager[1]	*TP*	*The Test Packager appears as separate component in the Smarter Balanced Architecture Report, but it is implemented as an integral part of Test Authoring because it shares all the data that is required to produce an assessment XML file that captures all the elements of a test and supplies all of the requirements of downstream components. Depending on the purpose of the test package, the package may include test items and stimuli in addition to a test XML file.*
Test Spec Bank	*TSB*	*The Test Spec Bank is a repository of approved and published tests. Downstream components browse the Test Spec Bank for completed test packages to download.*
Test Item Bank	*TIB*	*The Test Item Bank is repository of items and stimuli that are ready for field test or operational use. This is the source for item and stimuli information and content for all Test Authoring functions.*
Test Delivery System	*TDS*	*The Test Delivery System is an union of several components like Proctor and Student interface which when working together allow students to take a test.*
Core Standards	*CS*	*The Core Standards is an application which provides an interface to the Core Standard Repository database that is housing the Standards data.*

Table 1: Components Under Test

2 References

	Reference	Location	Version
1	Test Authoring User Guide	Test Authoring source code repository documentation folder	1.0
2	Task order 02 test cases	Test Authoring source code repository documentation folder	1.0
3	TO-2-3 Changes and Interface Specifications	Test Authoring source code repository documentation folder	1.0
4	Test Authoring Backlog Document	Test Authoring source code repository documentation folder	1.0

Table 2: References

3 Scope of testing

The main goal of Task Orders (TO) 2 and 3 is to ensure integration among all Test Authoring (TA)-related components. The full list of changes is described in "TO-2-3 Changes and Interface Specifications" and mostly involve changes in the XML packages generated by TA. Others affect REST API calls used by the associated components. There is also a list of specific changes made because of changes to the original requirements. Due to resource and time constraints, independent testing testing of each of these

[1] The functionality provided by Test Packager (TP) was integrated into Test Authoring during implementation of the original application. TP is listed here for reference only.

changes was unrealistic. Additionally, some changes were out of test scope since they required significant preparation efforts such as setting up a specific environment. For example, to verify outgoing API request, components must be reconfigured to send APIs to a specific test component capable of storing them.

The main priority of the test team was to ensure proper integration among components; this was tested by creating various types of tests and verifying that different student can take them. It is important to note that although the testers were familiar with components' functionality and the basic rules required to create a valid test, there are many rules which are not enforced by the application. Not following them could cause a test which looks legitimate from TA's perspective to function incorrectly in Test Delivery (TDS). Creating a real test using all available TA functionality is a complex task, which is performed by Test Authors who undergo specific training for that. So despite the variety of tests described in Section 5.3 , the testing for these task orders does not cover 100% of all TA functionality.

Changes made in TO 02 and 03 from testing perspective can be divided into two categories.
1) Changes that should be tested directly
2) Changes that are out of scope or changes which due to complexity and other reasons are better tested as a part of integration testing for the five individual components in Table 1.

This test plan will review all changes made for TO 02 and 03 and specify which of them belong to category (1). Scenarios for category (1) are described in the Test Cases Excel document. All other changes belong to category (2).

4 Task Order Change Categories

Changes from section 2.1.2 from TO-2-3 Changes and Interface Specifications document:

Category (1):

#1-18, #20, #23, #25, #29, #31-35, #37-48, #51

Category (2):

#19 is about configuration set up issue. TA does not validate any of these parameters; but incorrect parameters will cause issues in TDS.

#21 and #31 are about configuration set up. These parameters are located in Program Management (PM) and testing of those is out of scope.

#24 is documentation/set up issue, testing of this is out of scope

#26, #36, #51 are about outgoing API requests. Testing of these is validated via System Test.

All unspecified changes belong to category (2).

5 Integration testing

The purpose of this testing is to ensure that test created in TA can be taken by students.

5.1 Environment set up

Prior to the start of testing, the environment should be correctly set up including proper configuration in Program Management and data to be created/inserted in the following components:

1. User with create and publish tests permissions in TA and TSB should be added to Single Sign On (SSO) via Administration and Registration Tools (ART)
2. User with coordinator permissions for ART should be created in SSO via ART
3. User with proctor permissions for TDS should be created in SSO via ART
4. Entities such as state, district and school and students should be created in ART
5. Valid standards should be uploaded in Core Standards (CS)
6. Valid items should be uploaded to Test Item Bank (TIB)
7. Settings section of TA should be populated with Subject, Publication, Computational and Scoring rules information

5.2 Steps to create test

While the majority of data transactions between components happens automatically, some of them must be done manually in order to complete the test creation process. To create a test, the following steps must be followed:

1. Create and publish test in TA
2. Click request export for administration package in TSB for that test
3. Download zip file from SFTP and get the test specification XML from that zip
4. Open MySQL Workbench and connect to TDS
5. Execute query to insert test specification XML in to TDS, and modify test label if necessary
6. Open the Content Uploader app and upload items package from zip file, then click publish
7. Open ART and register the test, specifying eligibility in order to allow students to take that test

5.3 List of tests required for requirement coverage

The following is a list of tests with increasing complexity that should be created in a process of integration testing:

1. Test with one adaptive segment

2. Test with one fixed segment
3. Test with several adaptive and fixed segments
4. Test with several adaptive and fixed segments with additional items set up (changing items from OP to FT, specifying enemy items and groups and so on)
5. Same as previous test plus filled blueprint information
6. Same as previous test plus filled Affinity groups, Scoring, Reporting and Performance levels information
7. Next version of previous test with some changes
8. Copy of previous test with some changes

Note: The versioning functionality of TA used in steps 7 and 8 is not stable. Due to the issue described in backlog item SB-1466 (refer to Backlog Document), creating a new ***version*** of a multi-segmented test and uploading it to TDS may cause inconsistencies with that test and any previous instances of that test in TDS.