**My Form Builder**

## Scope

Automated assembly of test forms, items drawn from Intelitest, based on statistics and other beta data associated with item identifiers.

Benefits

By standardizing and automating existing work process we will

* Prevent errors in test construction – hitting the blueprint every time at the desired level of difficulty assures a valid and reliable test is delivered to the candidate
* Shorten time to market -- current process often takes clock time of a week or more
* Lower Labor costs
* Enhance / expand capabilities – with automation we will be able to offer multiple forms even with small bank clients to help prevent dis-honest test takers from cheating as well as expand revenue base

Note: Each of the above benefits needs to be measured under current process and have metrics of improvements defined before project begins.

Inputs into the process:

* Intelitest test specifications (e.g. client name, test title, etc)
* Content specs from test blueprint
* Target level of difficulty
* Recognize enemies and friend pairs in item selection
* Test/item history
* Equating information (to include designation of base form, and if form with be pre- or post-admin equated (affecting # of required equaters))
* Cut score (pre-equated)
* Test specific “rules”
  + Item re-use specifications based on form history (including % overlap allowed between forms)
  + Item state requirements (usable, pre-test, retired, in-use, etc.)
  + Rules for use of pre-test items
  + target # of items with graphics or other parameters
* Additional form requirements: (e.g. sets, items with graphic images, data from customized fields created and entered into Intelitest or the Item Metafile for this specific purpose)

Outputs

* Item identifiers of items to include on the test – i.e. list of proposed items for creating draft form
* Actual expected level of difficulty
* Actual content specifications

Additional process

* Test developer creates test form – or automated process creates form by drawing item content from Intelitest
* Test Developer reviews draft form (overlap check) and marks any required changes (For majority of clients this step will also follow with Exam Committee Review)
* Test developer manually chooses replacement items –we shall create a parameter driven item search tool which allows exclusion [what items are already on the form, prior forms, etc.] and inclusion [allow search by stat, content area, graphic included, and other parameters, etc.]
* Form identification added by test developer
* After final form review, content is passed on to publishing

## Technical Architecture

**Windows vs. Web Based Application**

### Windows:

**Pros:**

* Simple
* More Secure
* Faster Development

**Cons:**

* Cannot consume Web services
* Download and Installment required
* Administration local Machine access rights

### Web based

**Pros**

* Fewer risks
* Wider Community
* More Accessible
* Easier Deployment
* Easier updates

**Cons**

* Slower Development
* Slower Speed

### Design

**Option 1 – Application separate Entit**y

Login (Intelitest Credentials)

Form Builder Application

Business Logic

Form structure

Item codes

Intelitest

Web Services

Intelitest DB

Web UI

New form

Create

Add items to form

1 List of master codes

Darap and drop

Preview

Advanced Search

50 items that match Citeria

List

Preview

Darp and Drop

**Option 2 – Build as separate Web App entirely that uses Intelitest Web Services to load content**

Pros:

* Keeps development clean and separate from Intelitest
* Completely separate application. Not bound by current Intelitest structures and architecture
* Ability to make updates separate of Intelitest. Quicker and faster
* Form Structure can be stored in a separate relational DB

Cons:

* Would need to use class libraries from Intelitest for objects

Build into Intelitest

**Option 1 – Build within Intelitest Web App**

Pros:

* No separate deployment
* All libraries within Intelitest are directly accessible
* All functionality in one project location

Cons:

* Would need to use existing Intelitest design pattern and structure
* Would be new Intelitest functionality as opposed to a new application
* Intelitest already has form building functionality

Form Builder