Web-Based Assessment Data Collection: Using User-Centered Design and Agile Programming Methodologies to Ensure Usability and User Buy-In

2007 NCES MIS Conference  March 1, 2007  session VII-D

Notes:

All images contained herein created by APS/RDA and are copyright released for inclusion in this presentation file.

All data shown are fictitious and for demonstration purposes only. Any resemblance to real students is purely coincidental.
Web-Based Assessment Data Collection: Using User-Centered Design and Agile Programming Methodologies to Ensure Usability and User Buy-In

Shayne Kendall, Mark Leo-Russell & Terri Christiansen
Research, Development & Accountability
Albuquerque Public Schools
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
- Resources and References
Your Presenters

- **RDA Technology Team:**
  - **Shayne Kendall, Technology Manager**
    - B.B.A., M.I.S. Concentration, University of New Mexico
    - Employed by APS since 2002 – worked up through ranks from Technical Assistant to Systems Analyst to team Manager
    - Experience in database design and programming in health care and education sectors
  - **Mark Leo-Russell, Web Applications Developer**
    - B.S. Geology, New Mexico Institute of Mining & Technology
    - Current working on a M.A., Organizational Learning & Instructional Technologies (OLIT), University of New Mexico
    - Extensive experience in software development, systems engineering and information design
  - **Terri Christiansen, Database Administrator**
    - B.B.A., Business Administration, Eastern New Mexico University
    - Data Processing Certificate, Albuquerque Technical-Vocational Institute
    - M.I.S. experience in Government, Education and Private Sectors
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
Background and History

- Albuquerque Public Schools (APS)
  - 34th largest school district in nation
    - ~90,000 students at 130 schools
  - Ethnicity:
    - 52% Hispanic
    - 36% Caucasian
    - 2% Asian
    - 5% Native American
    - 4% African American
    - 1% other
  - Organized into 13 clusters
    - Based on geographic location within the city
    - Centered around high schools and their feeder schools
Background and History (cont’d)

- **APS Research Development & Accountability**
  - District department responsible for assessment, testing, state & federal accountability, reporting, research, and data quality assurance/integrity
  - 35 staff members organized into 6 teams:
    - Accountability (5)
    - Assessment (6)
    - Research & Evaluation (7)
    - Testing (6)
    - Technology (4)
    - Data Integrity (4)
Background and History (cont’d)

Recent evolution of RDA Technology and Web Applications

1998: Initial data entry system: information from schools collected using paper instruments then hand-entered into spreadsheets.


2001: Web and file server deployed at RDA; Suspension site released district-wide.

Benefits of web-based data handling reach critical mass in department and district; numerous requests for additional applications; analyst position added to staff.

2003: Staffing up to 3 analysts charged with web development and databases; 2 servers hosting 8 web applications and file services.

2005: Staff turnovers/promotions result in “new blood” – planning begins on “new era” of RDA web applications and database management.

2006: RDA IT Team evaluates technologies, development methodologies, IDEs, programming conventions, site structure. User requirements gathered thru focus groups, training sessions and presentations.

2007: RDA expands to take on district SIS data integrity and QA while supporting 6 servers and 14 web applications (plus in-house tech support).
Background and History (cont’d)

- RDA Technology Team

  Fourteen (14) total web applications + Server Maintenance + Database Administration + District/School Reporting + In-house Tech Support

  Three (3) Technical Staffers

  = LOTS OF WORK !!
Efficiency and accuracy are key!

- Our goal? Find a web platform that has the following:
  - Functionality that relates to the vision of the project
  - Graphing, built-in form validation, plethora of pre-built functions
  - Procedural and Object oriented (OOP) options
  - Cost effective to implement and maintain
  - Shortest possible learning curve
  - Scalability
  - Supports rapid application development (RAD)
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
System Architecture

- Currently undergoing multi-year project to completely redesign the RDA web site and associated applications:
  - Including look & feel, navigation and back-end
    - Server architecture, databases
  - Fundamental change in organization and structure, moving from separate independent applications to an integrated assessment management and performance support system.
System Architecture (cont’d)

- Current RDA Web Applications Design Model
System Architecture (cont’d)

- Future RDA Web Applications Design Model

**RDA EXTERNAL**

**Secure Area**
- Class lists based on Logon
- System looks at each student's grade and demographics to determine areas of access
- Secure/Private Documents
- Intranet for RDA Employees

**Guest/Public Area**
- Public Reports, Data and Information
System Architecture (cont’d)

- Future RDA Web Applications - Tiered Access
  - Multi-level access based on job position and/or needed data access permissions
  - Create, edit, delete, view, and report options at each level

District Administrators

Cluster-Level Staff
Cluster Level Principals, Curriculum Assistants, etc.

School Staff
Principals, Assistant Principals, Counselors, etc.

Grade-Level/Department-Level
Grade Chairs, Department Chairs

Class
Teachers, EAs, etc.
The life of a teacher (currently)

1. Logon using Employee Number and Access Code. Demographics are pulled from HR and payroll systems.
The life of a teacher (currently)

2. Choose the application from a list of approved apps.
The life of a teacher (currently)

3. Arrive at the application and enter student by student using the student ID. Enter assessment scores, interventions, etc. and submit. Repeat process for each student. Student information is pulled daily from SchoolMax district SIS.
The life of a teacher (future)

- Anything “public” will be accessible without logging on
- A class list for that teacher will be displayed, not a list of applications
- Each student will have access to only the web forms that correspond with that student’s demographics
- District, school and class level reports will be available immediately after testing windows. Disaggregated demographic data using tabular and graphing components
- Each student will be flagged as “tested or “not tested” during mandatory testing windows. This drastically reduces untested students
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
Technology Adoptions

- We examined and analyzed technology in two key areas before deciding new adoptions:
  
  - Web server software and associated development platform
  
  - Database management system
The verdict is in!

- After research it was between PHP, ASP.NET and ColdFusion
  - PHP was scratched:
    - Learning curve too long. No team member had experience in PHP
  - ASP.NET was considered except:
    - Object-Oriented Programming (OOP) experience needed immediately to do effective development, in-turn a long learning curve
    - Takes more code to achieve the same functionality that other languages can do in a couple lines. Not good for rapid development
  - Onward with ColdFusion:
    - It met all of our programming goals
    - Members of our team already had experience with ColdFusion.
    - Most bang for the buck, take a look at the following code.....
Example ASP (our original programming language) code calling a SQLServer stored procedure:

```vbscript
sub OpenDatabase()
Const adExecuteNoRecords = 128
Dim cnnStoredProc ' Connection object, cmdStoredProc ' Command object
Dim rstStoredProc ' Recordset object (for part 2), paramId, paramId2, paramId3, var_error, var_redirect
Set cnnStoredProc = Server.CreateObject("ADODB.Connection")
cnnStoredProc.Open "DSN=Math"
Set cmdStoredProc = Server.CreateObject("ADODB.Command")
cmdStoredProc.ActiveConnection = cnnStoredProc
cmdStoredProc.CommandText = "usp_Math_Validate_Student"
cmdStoredProc.CommandType = adCmdStoredProc
Set paramId = cmdStoredProc.CreateParameter("@enteredIDNumber", adInteger, adParamInput)
paramId.Value = Session("sesStudentID")
cmdStoredProc.Parameters.Append paramId
Set paramId2 = cmdStoredProc.CreateParameter("@locationNumber", adInteger, adParamInput)
paramId2.Value = Session.Contents("SesLocationNumber")
cmdStoredProc.Parameters.Append paramId2
Set paramId3 = cmdStoredProc.CreateParameter("@error", adVarChar, adParamOutput, 255)
paramId3.Value = 
""
cmdStoredProc.Parameters.Append paramId3
cmdStoredProc.Execute , , adExecuteNoRecords
var_error = cmdStoredProc.Parameters("@error").Value
Set rstStoredProc = cmdStoredProc.Execute
```

2007 NCES MIS Conference [March 1, 2007, session VII-D]
Example ColdFusion code calling a similar SQLSERVER stored procedure:

```coldfusion
<cfstoredproc procedure="usp_ELLWaiverRequest_Insert" datasource="ELLWAIVERJDBC" returncode="yes">
  <cfprocparam type="In" cfsqltype="CF_SQL_NUMERIC" dbvarname="@State_ID" value="#Session.Student.StateID#" null="no">
  <cfprocparam type="In" cfsqltype="cf_sql_varchar" dbvarname="@SAT_Members" value="#FORM.txt_SATMembers#" null="no">
  <cfprocparam type="Out" cfsqltype="cf_sql_varchar" variable="error" value="%error%" dbvarname="@error" maxlength="255">
</cfstoredproc>

• 19 less lines!
• 24 lines x 4000 web files = 96,000 lines of just procedure calls in ASP
• 5 lines x 4000 web files = 20,000 lines of just procedure calls in ColdFusion
• That’s 76,000 lines of code we don’t have to write in procedure calls!
Recapping…

- Summary: you’ve seen our organization, some background info, project history, architecture, and technology….

- Now on to our data and databases…
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
Data and Databases

- Obtaining student data from district’s Student Information System (SIS)
  - Old system: manual extracts from legacy system
  - 40-, 80-, 120-day and end-of-year downloads
  - Latency: ~40 days

- New system: daily automatic sync between district SIS and RDA database server
  - Latency: ~24 hours
Data and Databases (cont’d)

- Database organization
  - Old system: multiple Access databases
  - No standards for schemas, data types, etc.
  - New system: SQL Server
    - Standardized schemas, data types, etc.
    - Data “pulled in” from numerous databases from other APS depart
      - SchoolMax SIS, HR, payroll, outside vendors
Database Organization

Old system:
- Multiple Access MDB files feeding web applications
- Some database files contain same or similar data tables (e.g., school location information)
Database Organization

New system:
- Single SQL Server feeding web applications
- One consolidated copy of common data tables (e.g., school location information)
Data and Databases (cont’d)

- Migration from Access to SQL Server. Why?
  - Scalability, Speed, Flexibility
  - Stored Procedures
  - Normalization and Consolidation
# Access-2000 vs. SQL Server-2000

<table>
<thead>
<tr>
<th>Object</th>
<th>Maximum sizes/numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database size</td>
<td>1 Gb</td>
</tr>
<tr>
<td>Number of characters in an object name</td>
<td>64</td>
</tr>
<tr>
<td>Number of characters in a password</td>
<td>14</td>
</tr>
<tr>
<td>Number of characters in a user name or group name</td>
<td>20</td>
</tr>
<tr>
<td>Number of concurrent users</td>
<td>255</td>
</tr>
<tr>
<td>Number of characters in a table name</td>
<td>64</td>
</tr>
<tr>
<td>Number of characters in a field name</td>
<td>64</td>
</tr>
<tr>
<td>Number of fields in a table</td>
<td>255</td>
</tr>
<tr>
<td>Number of characters in a Text field</td>
<td>255</td>
</tr>
<tr>
<td>Number of characters in a Memo field</td>
<td>65,535 / 1 Gb</td>
</tr>
</tbody>
</table>

Source:  
http://www.mssqlcity.com/Articles/Compare/SQLvsAccess.htm  
2007 NCES MIS Conference [March 1, 2007, session VII-D]
### Access-2000 vs. SQL Server-2000

<table>
<thead>
<tr>
<th>Object</th>
<th>Maximum sizes/numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>65,536 * Network Packet Size</td>
</tr>
<tr>
<td>Bytes per short string column</td>
<td>8,000</td>
</tr>
<tr>
<td>Bytes per text, ntext, or image column</td>
<td>2 GB-2</td>
</tr>
<tr>
<td>Bytes per index</td>
<td>900</td>
</tr>
<tr>
<td>Bytes per foreign key</td>
<td>900</td>
</tr>
<tr>
<td>Bytes per primary key</td>
<td>900</td>
</tr>
<tr>
<td>Bytes per row</td>
<td>8,060</td>
</tr>
<tr>
<td>Bytes in source text of a stored procedure</td>
<td>Lesser of batch size or 250 MB</td>
</tr>
<tr>
<td>Clustered indexes per table</td>
<td>1</td>
</tr>
<tr>
<td>Columns per index</td>
<td>16</td>
</tr>
<tr>
<td>Columns per foreign key</td>
<td>16</td>
</tr>
<tr>
<td>Columns per primary key</td>
<td>16</td>
</tr>
<tr>
<td>Columns per base table</td>
<td>1,024</td>
</tr>
<tr>
<td>Columns per SELECT statement</td>
<td>4,096</td>
</tr>
<tr>
<td>Columns per INSERT statement</td>
<td>1,024</td>
</tr>
<tr>
<td>Connections per client</td>
<td>Maximum value of configured connections</td>
</tr>
<tr>
<td>Database size</td>
<td>1,048,516 TB</td>
</tr>
<tr>
<td>Databases per instance of SQL Server</td>
<td>32,767</td>
</tr>
</tbody>
</table>
## Access-2000 vs. SQL Server-2000

<table>
<thead>
<tr>
<th>Object</th>
<th>Maximum sizes/numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filegroups per database</td>
<td>256</td>
</tr>
<tr>
<td>Files per database</td>
<td>32,767</td>
</tr>
<tr>
<td>File size (data)</td>
<td>32 TB</td>
</tr>
<tr>
<td>Identifier length (in characters)</td>
<td>128</td>
</tr>
<tr>
<td><strong>Locks per connection</strong></td>
<td><strong>Max. locks per server</strong></td>
</tr>
<tr>
<td>Nested stored procedure levels</td>
<td>32</td>
</tr>
<tr>
<td>Nested subqueries</td>
<td>32</td>
</tr>
<tr>
<td>Nested trigger levels</td>
<td>32</td>
</tr>
<tr>
<td>Nonclustered indexes per table</td>
<td>249</td>
</tr>
<tr>
<td>Objects in a database</td>
<td>2,147,483,6474</td>
</tr>
<tr>
<td>Parameters per stored procedure</td>
<td>1,024</td>
</tr>
<tr>
<td>REFERENCES per table</td>
<td>253</td>
</tr>
<tr>
<td>Rows per table</td>
<td>Limited by available storage</td>
</tr>
<tr>
<td>Tables per database</td>
<td>Limited by number of objects in a database</td>
</tr>
<tr>
<td>Tables per SELECT statement</td>
<td>256</td>
</tr>
<tr>
<td>Triggers per table</td>
<td>Limited by number of objects in a database</td>
</tr>
<tr>
<td>UNIQUE indexes or constraints per table</td>
<td>249 nonclustered and 1 clustered</td>
</tr>
</tbody>
</table>
## Access-2000 vs. SQL Server-2000

<table>
<thead>
<tr>
<th>Object</th>
<th>Access</th>
<th>SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database size</td>
<td>1 Gb</td>
<td>1,048,516 Tb</td>
</tr>
<tr>
<td>Number of concurrent users</td>
<td>255</td>
<td>Maximum value of configured connections</td>
</tr>
<tr>
<td>Number of fields in a table</td>
<td>255</td>
<td>1,024</td>
</tr>
</tbody>
</table>
Recapping again…

- Summary: you’ve seen some background information, our architecture, technology adoptions and the databases….

- Now let’s look at the people side of the equation….
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
Development Methodologies

- In addition to a new site architecture, the move to ColdFusion, and the implementation of SQL Server, we are embracing modern software development models:
  - User Centered Design (UCD)
  - Agile programming
Development Methodologies

- User Centered Design (UCD)
  - Design philosophy and processes where the needs, wants and requirements of users drive development efforts
  - In short, systems are designed and constructed from the users’ perspective, not the programmers’ perspective
Development Methodologies

- User Centered Design – Advantages:
  - User buy-in
  - Community sense of ownership
  - Tools created by and for the customers
    - In our case: teachers, principals, district staff, etc.
  - Users have as much invested in the development project as the software team
Development Methodologies

- User Centered Design – Advantages (cont’d):
  - Users guide the priorities of development – what features or capabilities are important to THEM at THIS POINT IN TIME
  - The software evolves as the organization changes
    - ex: changes in an assessment process or business rule are immediately reflected by user-directed changes in the associated web site
Development Methodologies

- User Centered Design - Techniques:
  - Focus Groups
  - Usability Testing
  - Presentations and Training
    - Feedback from users is collected during non-technical sessions such as assessment or testing training
  - “PR” within the district
    - ex: APS’s recent TechConnections day
  - Conversations with staff at all levels of the school district
Development Methodologies

- User Centered Design - Recent work by our team:
  - Reading Program Inventory
    - Phase 1 Fall 2006: designed, developed and deployed first version – data being analyzed now
    - Phase 2 Spring 2007: revised on-line questionnaire to be released in the next week or so
  - KIDS/KDPR (APS Kindergarten assessment)
    - First version developed last summer and used this past fall; eliminate bubble sheets and greatly reduced data report time
    - Second round of focus groups with K teachers in December
    - Some changes to be incorporated this spring with more user-requested enhancements to be added for fall 2007
Development Methodologies

- Agile Programming
  - Software engineering conceptual framework that values:
    - Individuals and interactions over processes and tools
    - Working software over comprehensive documentation
    - Customer collaboration over contract negotiation
    - Responding to change over following a plan
Development Methodologies

- Agile Programming vs “Traditional” Development

- Traditional Development (sometimes known as “Waterfall”)
  - Long cycles of requirements-development-testing
  - Typical cycle time: months to years
  - Formal specifications and documentation

- Agile Development
  - Short continuous iterations of development ("time boxes")
  - Typical cycle time: weeks to months (in some cases, days)
  - Informal specifications and documentation
Development Methodologies

- Agile Programming and User-Center Design
  - The constant iterative nature of Agile development dovetails into the continuous communications of User Center design techniques, especially as related to what users consider important in the current development cycle.
  - Development team strives to deliver small sets of features and program capabilities that users want or need NOW
    - Especially relevant to education IT given ever-changing assessment practices, reporting regulations and the like
Presentation Overview

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
Example Applications

- Web-Based Assessment Tools:
  - KIDS – KDPR Information & Diagnostics System
    - KDPR – Kindergarten Developmental Progress Report
    - District-designed short-cycle observational assessment
  - AIP – Academic Improvement Plan
    - Used to monitor and report reading and math assessments and interventions
Example Application

KIDS

Main application web page showing links for data entry, reports, and assessment-related resources.
Example Application

KIDS (cont’d)

Top portion of the data entry/edit web page for assessment items.
Examples of class-level report web pages: assessment status (left) and assessment summaries (right).
Example Application

KIDS (cont’d)

Portion of the individual student assessment report web page.
Example Application

AIP

Main application web page showing links for data entry, reports, and assessment-related resources. Note status flags in left column.
Example Application

AIP (cont'd)

Top portion of the data entry/edit web page for Academic Improvement Plan items.
Example Application

AIP (cont’d)

Bottom portion of the data entry/edit web page for Academic Improvement Plan items.
Presentation Review

- Background and History
- System Architecture
- Technology Adoptions
- Data and Databases
- Development Methodologies
- Example Applications
- Summary and Closing
The End
To download our presentation, point your browser to:

- rda.aps.edu

For further information, contact the APS RDA Technology Team:

- rdatechsupport@aps.edu
Resources & References

The following books and web sites will give you more information about the topics covered in this presentation.

1. User Centered Design & Usability
2. Agile Development
3. Programming & Software Development
4. ColdFusion
Resources and References

1. User Centered Design & Usability

- Don’t Make Me Think: A Common Sense Approach to Web Usability (2nd Edition)
  - Steve Krug
  - ISBN: 0321344758

- Designing Web Usability: The Practice of Simplicity
  - Jakob Nielsen
  - ISBN: 156205810X

- Prioritizing Web Usability
  - Jakob Nielsen & Hoa Loranger
  - ISBN: 0321350316
Resources and References

1. User Centered Design & Usability (cont’d)

   • useit.com: Jakob Nielsen's Website
     • www.useit.com

   • Usability.gov - your guide to developing usable and useful web sites
     • www.usability.gov
Resources and References

2. Agile Development

Practices of an Agile Developer
Venkat Subramaniam & Andy Hunt
ISBN: 097451408X

Agile and Iterative Development – A Manager’s Guide
Craig Larman
ISBN: 0131111558

Agile Project Management
Jim Highsmith
ISBN: 0321219775

Agile Database Techniques – Effective Strategies for the Agile Software Developer
Scott Ambler
ISBN: 0471202835
Resources and References

2. Agile Development (cont’d)

• Manifesto for Agile Software Development
  • www.agilemanifesto.org

• Agile Alliance
  • www.agilealliance.org
Resources and References

3. Programming & Software Development

The Pragmatic Programmer – from journeyman to master
Andrew Hunt & David Thomas
ISBN: 020161622X

Code Complete – A practical handbook of software construction
Steve McConnell
ISBN: 0735619670
Resources and References

4. ColdFusion

ColdFusion MX7 Web Application Construction Kit
Ben Forta, Raymond Camden,
Leon Chalnick & Angela C. Buraglia
ISBN:0321223675
includes CD with Developer version of ColdFusion MX7 and additional materials
Resources and References

4. ColdFusion (cont’d)

- Adobe's ColdFusion product page
  - www.adobe.com/products/coldfusion
  - Site includes trial versions, product demonstrations, developer references, library of CF code and custom tags