

FINAL - Replatforming/Refactoring Vendor Presentations

**NCCSD Systems/Data Analysis Workgroup
Systems Modernization Vendor Webinars Oct. 18/Oct. 25/Nov. 1, 2019
“Replatforming/Refactoring”(R/R) and “Low Code/COTS” (LC/COTS) Approaches
Recap and Most Interesting Lessons/Similarities from the Vendors**

Caution: This document contains information presented at confidential sessions between specific vendors and state representatives and may contain information that could be proprietary. Please consider this when sharing with non-State or your vendor staff.

5 Participating Vendors (in alphabetical order):

Accenture, Cambria Solutions, Conduent, Deloitte, Protech

Terminology/Definitions:

These presentations confirmed that the terms are not at all consistently defined and/or used (see answers to question 1 on the Vendor Q&A documents, their PowerPoint presentations, and the presentation material at conferences such as ISM or WICSEC)

Other terminology cited by the vendors: “lift and shift” or “rehosting” for replatforming and “code conversion” or “language migration” for refactoring. From Deloitte: “One additional term that is often used is migration. This is generally a combination of replatforming and refactoring, moving an application entirely from a legacy platform, codebase, and database (e.g. Cobol on DB2/CICS Mainframe) to a modern platform, codebase, and database (e.g. Java on DB2/Linux/Cloud). Automated and manual variations exist.”

Approaches:

Basic concepts without worrying too much about terminology:

Replatforming usually refers to moving to a more modern technology platform. Vendor examples were moving from DB2/CICS on a mainframe to DB2/Oracle on Windows/Linux or more generically from a mainframe platform to a Java or .NET environment. They pointed out that in order to achieve this move, code conversion (refactoring) also may need to occur. An example from Conduent is as follows: “For example, high volume system processing in batch programs may not achieve the required performance when converted and deployed on a new platform. These programs and processes may need to be changed or redesigned to obtain faster processing on the new platform.” From Deloitte: “In the context of migrating legacy applications, refactoring generally means recreating legacy code in a modern coding language, generally for the purpose of enabling a replatform and improved maintainability.”

Refactoring can also be a standalone project, but can take more time and effort, because in order for most child support systems to move to a more modern language the project is not your typical line-by-line “clean-up”. As Accenture described in their document “....converting a largely procedural language, like COBOL, to an object-oriented language, like Java does not re-engineer the code to be optimized in this new paradigm of coding principles. It is only through converting, but then re-designing, or re-factoring that the language can be optimized for the more modern, object-oriented language.” (emphasis added) From Protech: “Conversion doesn’t always translate functional code into functional

code (like translating Greek to Latin, one language may not have an exact match in the other), so code has to be tested and sometimes rewritten. Additionally, libraries, functions, parameters and coding standards may be different requiring meticulous manual focus to provide the same functionality while invoking current libraries and functions and complying with modern coding standards.”

Major takeaways and consistent themes from the sessions:

- Neither of these approaches changes/improves the business functionality – they are simply a way to move to modern technology. Screens look very similar between old and new. Conduent and Deloitte gave us the best “before” and “after” screen shot examples – see below.
- The biggest organizational change impact is on the state’s IT staff – infrastructure tasks may be different, but existing legacy programmers may have an easier time transitioning to the new technology. Finding IT resources in general should be easier.
- With changes to code (refactoring), much discussion centered on how automated the conversion could be. The percentage of code that their tools can automatically convert drives timeframe and cost. One vendor suggested that states consider making existing code available to vendors in the procurement in order to get more accurate proposals. Another discussion point was whether the new code would be “full” Java or .Net, or a “lighter” version. The latter can require more refinement.
- The R/R approach may be best for states with more limited budgets who are happy with their existing functionality, or as a first step for states who need to modernize the technology relatively quickly, and then concentrate on functionality improvements. Cambria Solutions gave us an example from Mississippi of taking this approach and then “bolting on” other more modern auxiliary tools such as chatbot technology.
- This approach also takes strong planning and communication between the technology groups, and can’t be done well without strong business awareness and involvement -- even though at first glance it might appear to be a “technology only” project.
- This approach is also sometimes taken when a state wants to modernize all the human services systems controlled by one umbrella agency relatively quickly in an enterprise approach – moving them all to the new technology and then deciding on other improvements on an individual system by system basis.

Timeframes and Cost.

Timeframe consensus seemed to be 12-24 months.

Vendor Cost, when answered, ranged from \$500K to 15M, depending on which approach.

All vendors cited multiple other factors as influencing both timeline and cost, including:

- Whether database is being migrated
- Percentage of automated conversion achieved
- Number and complexity of screens, lines of code etc.
- Type of statewide rollout (phased or statewide)
- Other 3rd party products integrated within the system that might be affected
- Integration touchpoints with other systems

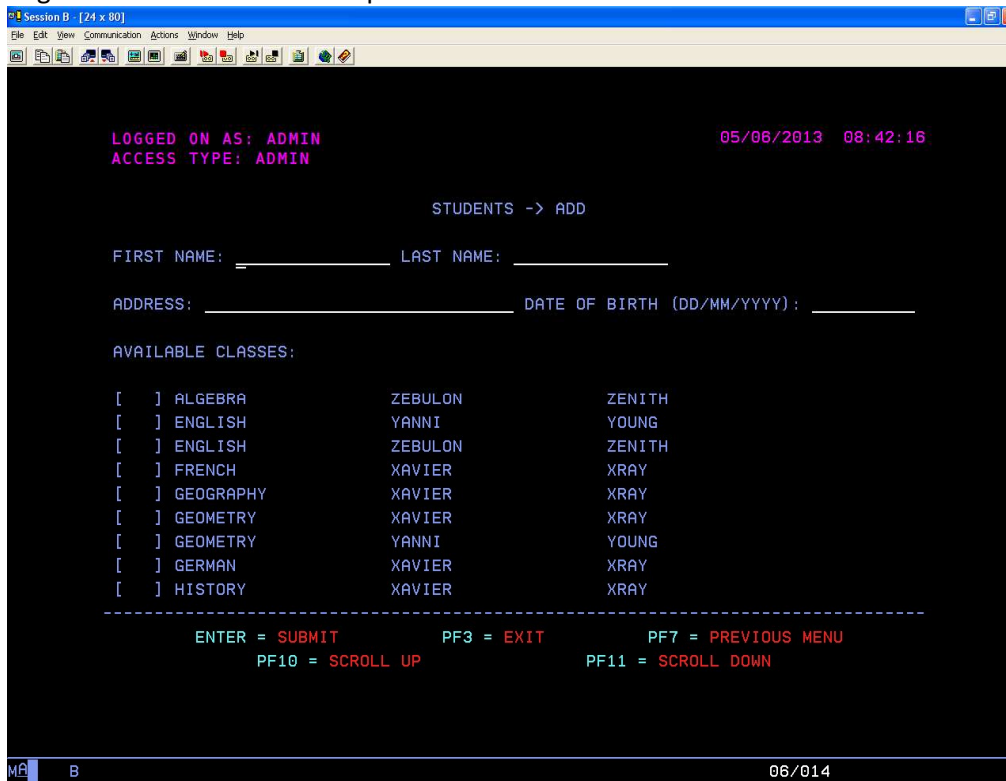
Miscellaneous:

Agile methodology works well with this approach. Mississippi used a Kanban approach.

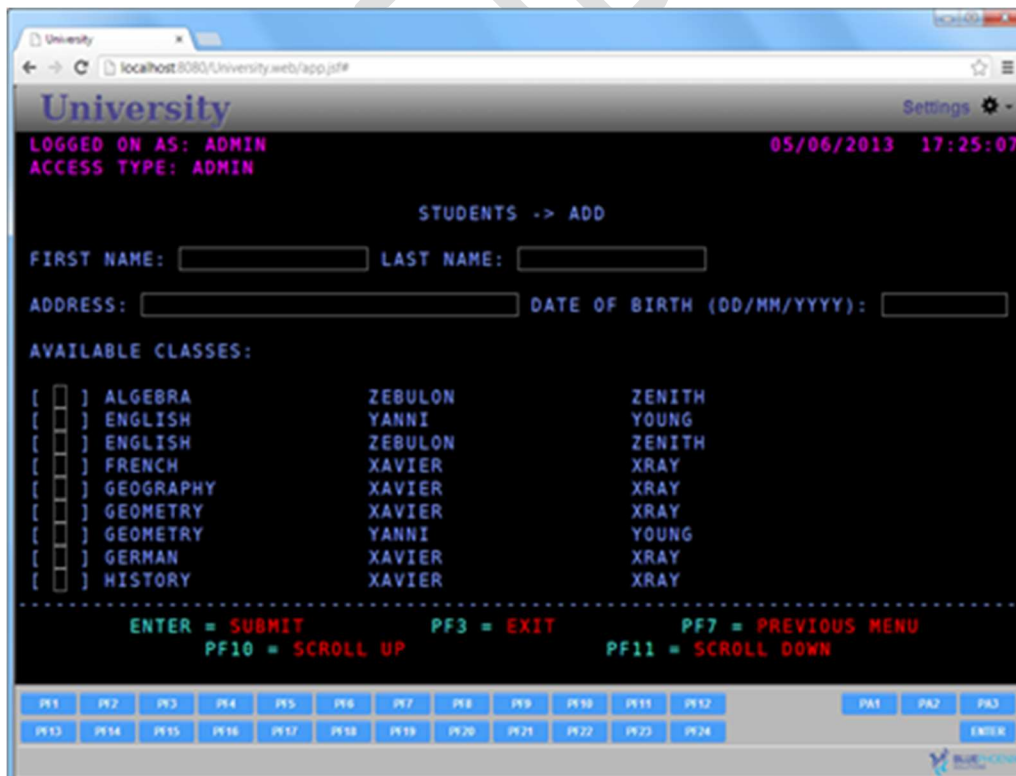
Before and After Examples:

From Conduent:

The following are examples of a mainframe screen showing the before and after migration to a browser-based platform.



BEFORE: Screen example of a mainframe screen before re-platforming



AFTER: Screen example after re-platforming to a browser-based platform

From Deloitte:
 Below are samples from a DMV refactoring project.

Legacy Screen



Source: Texas DMV

Migrated Screen



Source: Texas DMV

CONFIDENTIAL