NCCSD Systems Workgroup Vendor Forums – Q&A related to Refactoring/Replatforming

Vendor Name: Cambria Solutions, Inc.

Please enter your responses into this document, but feel free to send any other attachments as well.

Questions:

1. Please explain how your company defines both re-platforming and refactoring.

   Opportunities to reduce IT spend and increase architectural flexibility across an organization are at an all-time high, from the advent of Cloud computing, to easily transitioning legacy workloads to new platforms, and to implementing new modern platforms such as Low-Code/COTS. Cambria Solutions, Inc. (Cambria) has partnered with more than 10 states and large municipalities by strategizing, planning, and implementing solutions that help reduce costs and increase flexibility. Recently completing one of the largest re-platforms in the country (five systems concurrently), we have seen first-hand the promise of these approaches. In our experience, migrating rapid, low-risk mainframes to distributed environments are becoming the norm. Thousands of Microprocessor without Interlocked Pipeline Stages (MIPS) are being moved to modernized X86 platforms. They are running on Windows and Linux through low risk and rapid migration projects, and mainframes are being decommissioned with accelerating frequency. Although these migrations are becoming increasingly common, rapid, and low risk, many states and counties are weighing modernization as well. Of the many critical success factors for these projects such as strong project management, product ownership, and solution architecture, the most important and first step is to determine your approach and the correct nomenclature.

   In our methodology and through our experience delivering these projects, here are our definitions:

   - ***Re-platforming*** is moving the application code and data persistence layer of the programs over to a platform that supports it without modifying code and only changing the operating system layer and underlying infrastructure. As an example, if applications are written in Natural and ADABAS, then re-platforming moves that same Natural and ADABAS code onto a new operating system such as LINUX. Some code rewrite may be necessary based on the assessment, but it is limited.
   - ***Refactoring*** is converting the code from one code base to another while changing the functionality. In this example, Natural and ADABAS code are converted to a new language such as Java and the data persistence layer changed to some relational or NoSQL platform. Comparatively, this is the most hands-on approach and is lengthier than re-platforming. It creates more delivery risk as you are changing code bases and re-engineering applications while re-platforming to a new infrastructure involves fewer moving parts.

2. Are you able to share any screen shots of a "before" and "after" implementation of this approach?

   This is one of the key concepts in a re-platforming approach. When the applications are re-platformed, there is no change in the “before” and “after” presentation of the application. The

screens, operations and functions work exactly as they had on the mainframe infrastructure. Terminal emulators, such as Reflections, continue to function in the same way except for emulating X11 instead of the traditional 3270 protocol. In our experience with this approach, our team has executed the migration over the weekend and the workers came to work on Monday and begin using the application as if nothing happened. The amount of disruption and change to the organization is minimal and the benefits of no longer supporting the large mainframe infrastructure is great.

For the refactoring approach, it can produce varied results depending on the user interface (UI) framework adopted and the target language chosen. The UI can be fully modernized using Bootstrap libraries and Java or an approach using Microsoft technologies. Although the refactoring automation can do basic rendering of a new UI screen based on the legacy screen, there is significant re-work required to properly position and implement a usable interface to the worker. However, the worker will eventually receive a more modern, enhanced UI in the target language. Note: We can provide example screenshots of both approaches upon request.

3. Under what circumstances does it make the best sense for a state child support agency to consider refactoring/replatforming versus other possible means of modernizing its child support system? Are there any characteristics of either a state's IT system or its business processes that lend themselves more to this approach?

The operational priority of the agency should drive the approach. If the priority is to unshackle the agency from legacy hardware and the maintenance costs that come with them without disrupting the current operational processes and workforce, then re-platforming can be an attractive approach as it is low-cost, low-risk and can be done quickly.

Refactoring should be considered if the organization has a strong IT organization and is interested in building and maintaining their own technology. Although the transition can be made to a modernized codebase, the implementation and operational maintenance of the system requires a hands-on IT organization and good product development practices to continue to maintain and improve on the refactored solution. The Agency must also consider the availability of high-quality developers such as those that can develop Java or .NET/C# code to support the refactored solution.

Modernization of the system either through a complete code re-write or the implementation of a low-code solution can be a significant effort, but with large benefits. The amount of change to the organization is significant from business process all the way through the IT support organization. If the Agency is in the process of consolidating or refactoring business processes where staff will be significantly affected, it may be a good approach to modernize the technology systems in conjunction with this effort. Below is a high-level summary of the circumstances, budget, and time for each approach:

Approach	Circumstance Summary	Budget	Time
Re-platform	System or software is out of support, or too costly to support for infrastructure. The Agency Desire is to not affect the current business processes or staff too much. The available budget is low.	Lowest	Shortest (<12 months)

Refactor	The current system has a clean and consistent codebase with the ability to support a new code base. There is interest in retooling current business processes and in modernizing staff skills.	Medium	Shorter (18-24 months)
Modernization	The Agency is undergoing a complete business area reengineering effort with new business processes and retrained staff. IT Organization is skilled in the latest technologies and has a desire for product management of the system. The budget available is sufficient for a large-scale replacement.	Medium-High	Long (24 months+)

4. Generally speaking, what should a state expect on the following: project timeframe, project cost, time to rollout statewide?

There are several factors that go into both the schedule and cost of a re-platforming project. A range can be as little as $500k an application up to $2M per application. The estimate is highly dependent on the quality of the code, number of libraries, or batch jobs as well as the preparation of the data. The best approach in re-platforming is to have a low-cost assessment done of the current applications. In our previous work with multiple states, we always start with the assessment and produce a cost benefit analysis to demonstrate the return on investment and finalize the cost of the potential project.

This assessment can evaluate number of code lines, proprietary properties of data such as packed or binary which must be adjusted when converted for the new platform and general code quality which can determine the amount of manual adjustments that must be made prior to conversion. Finally, one of the best aspects of re-platforming on the timeline is that it has not required an Implementation Advance Planning Document (IAPD) since you are not converting the codebase. This reduces an Agency's timeframe by many months. For a typical child support application, the average timeline could be less than 12 months.

Refactoring, as we have discussed, is highly dependent on the quality of the source code. Using refactoring tools in some cases 90% of the code can be factored in other cases as little as 20% can be refactored leaving the balance to be manually re-written. This effort and cost can be significant and sometimes yields completely scrapping and re-writing the base application or looking for an alternative approach for that system. For a typical child support application, the range could be $2M to $4M+. The average timeline could be 18 to 24 months.

5. Please list and explain the pros and cons, and any common pitfalls the states should know, for refactoring/re-platforming. What surprised you in your implementations?

For re-platforming, one pitfall is that the Agency needs to be prepared to take on infrastructure tasks that were most likely being done by their state data center or statewide IT group. In our experience, while the Agency has maintained the code and supported their systems, they have not owned or fully operated the infrastructure surrounding those on the mainframe. When the systems are re-platformed, many of the tasks that the infrastructure team did on the mainframe still exist but now are just different tasks on the new platform and someone at the Agency will need to own them. Especially in the case of moving those workloads to the cloud, areas such as Linux administration, backups, and security administration may now be in the scope of the Agency IT organization.

A positive aspect of re-platforming is that the application is now available and open to interact easier with other non-mainframe systems that the Agency already has in its enterprise. Another positive aspect is that just being on the new platform will expose the Agency IT staff to new ways of doing things and managing changes that will transfer into future modernizations or replacement systems (e.g. DevOps).

A negative aspect to re-platforming is that overcoming the organizational change factors within the Agencies IT staff alone can seem overwhelming. For this to be truly successful, it takes strong leadership, constant messaging long before the project, and throughout the project about the benefits to both the Agency and the IT staff. It will require training and heavy hands-on interaction with the code and new processes throughout the project to give the Agency IT staff the comfort they will need to know that they are going to be able to support the system on the new platform.

6. What are the most important things that a state should do to prepare for this approach?

Consolidating code into one or two languages versus many will make the effort more simple and cleaner. Removing dead code also helps. In addition, making sure that you have all the source code for the objects you are running as well as making sure that all code being run for production is in your production libraries. For a re-platform, the effort is highly concentrated on lifting and shifting application functionality so when it is time for User Acceptance Testing the Agency needs to be ready to test the functionality of the system and compare it to production. For example, if a function does not work properly in production then it should not be expected that it will work properly just because the platform has changed. In a re-platform, the vendor has a high dependency on the Agency providing information about the system and its related interfaces.

Preparing your IT organization to administer the new platforms is also important. Regardless of the re-platforming or refactoring approach, the base infrastructure will be significantly different. Utilizing revised operating systems and potentially new security postures can be a significant shift from traditional legacy system management. Add in new code management approaches and deployment procedures and the IT organization in either approach can face the most challenging changes to their work.

7. How does this type of child support system modernization effort fit with states who need to have an enterprise approach?

Re-platforming supports an enterprise approach by creating the infrastructure to support the incremental modernization of solutions without significant disruption. Once the application is re-

platformed or refactored, modernization can begin. In our experience when re-platforming and designing infrastructure for the applications, we take an Enterprise Architecture approach planning for the next step of modernization as we deprecate the mainframe infrastructure. For example, in one state where we replatformed the child support system, we married the project with investments in a service-oriented architecture (SOA) with an Enterprise Service Bus, lean-agile transformation, and the deployment of a self-service bot (Gen). The Office of Child Support was able to gain the efficiencies from the re-platformed infrastructure while also advancing the enterprise architecture of the agency.

Our team considers the resource groups and technology that will be required to make legacy application data and functions to be exposed as services and an integration platform to enable the modular modernization of system functions in an incremental approach.

In addition – we look across the landscape of non-mainframe systems and determine how they fit into the overall topology of the Enterprise Architecture and promote re-use where possible and don't add in redundant componentry where capabilities already exist. An example would be if the Agency already has an Enterprise Analytics and Reporting approach, we consider how the re-platformed systems can leverage that existing functionality without adding addition tooling to support that function.

8. What haven't we asked that we should have?

What are the effects in each approach on the integration strategy for the Agency and how does it affect any external stakeholder systems that may interface with the Agency applications?