**NCCSD Systems Workgroup Vendor Forums – Q&A related to “Low Code/COTS”**

Vendor Name: Accenture

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

Questions:

1. Since there is not yet a consistent term or definition for this approach, please give your company's description, including your terminology and definitions. How is this approach different from a "custom" build of a child support system? If you choose to do a quick demo or screen shots that would be welcome.

**Low Code:** This term refers to using a platform, like Salesforce or Microsoft Dynamics, for system modernization. These platforms provide core CRM and/or ERP functions out of the box, but also a framework allowing the customer to build out additional functions and features using “configuration” rather than traditional code (i.e. programming). The platforms also offer a programming language so that unique requirements can be built via customization (i.e. coding) when the capability cannot otherwise be achieved. Hence, the term “low code” is used to emphasize the principle of configuration before code, or “clicks not code” when building a solution using these frameworks.

Further, the platforms offer an ecosystem where 3rd party providers can build out new capabilities and offer them as add-on products to enhance the platform’s native capabilities. Platform customers can leverage these add on products (freely or paid depending on the specific license) rather than customize or code the needed capabilities not supported by the Platform itself.

These platform solutions execute in the cloud and customers use the platform as a service (i.e. “pay as you go”) rather than procuring the platform as a software product. We specifically refer to this system modernization approach as “Platform as a Service” (PaaS). The low code/configuration framework and subscription pricing model differentiate this approach from a custom build.

**COTS**: This term is an acronym which stands for “Commercial Off the Shelf” product. It can represent any 3rd party software product which is commercially available. The term COTS distinguishes software products which are purchased from those that are custom built. Oracle, IBM, and Adobe software are examples of COTS products. Platforms like Salesforce and Microsoft Dynamics are also considered COTS, but are not traditional COTS as they are purchased as a service, not a product.

1. With reference to the "core" functionality required by the OCSE Systems Certification Guide (Case Initiation, Locate, Establishment, Case Management, Enforcement, and Financial Management), how does this approach handle each area? In particular, since Child Support requires complicated financial processing, e.g. distribution rules and arrears calculations, please address how these are handled with this approach.

The PaaS options available today originated from CRM, ERP, or other non CSE solutions. They adequately address most Child Support Case Management requirements with their native features, but do not offer “out-of-the-box” features to satisfy Locate, Interstate, Establishment, Enforcement, or Financial Requirements. These platform solutions will require both configuration, customization (i.e. code), and/or integration with other systems to provide processing which satisfies the “core” OCSE System requirements.

There are distinct architecture approaches for implementing CSE Solutions using Platforms. We use the following terms to differentiate two common approaches.

**Platform Integration –** This approach leverages the Platform as the User Interface, and integrates with a separate backend Child Support Engine implementation for core CSE functionality and Batch processing. The core CSE engine may be a Java or .Net code base converted from the State’s legacy system, a transfer system from another State, a custom-built solution, or a vendor’s CSE asset. (see slide 3 in the attached powerpoint file)

**All-In Platform –** In this approach all core CSE functionality is implemented directly in the Platform using configuration / customization. Integration is only used for peripheral functionality (e.g. ECM, Form Generation, Reporting, MDM, etc.) and external interfaces. (see slide 4 in the attached powerpoint file)

For implementing core CSE functionality, we recommend the “All-In Platform” approach, as “Platform Integration” requires complex data synchronization across multiple databases, duplicates business logic across the platform and integrated system, and relies heavily on web services which degrades performance.

1. What COTS or other products are used in conjunction with this approach to give a state a fully functional system?

The following are examples of peripheral functionality which can be provided through integration with COTS products. The COTS products listed by name are ones we have used recently on CSE implementations, but do not represent an exhaustive list of options.

* Form design and generation - Adobe Experience Manager Forms
* Reporting - SAS, Cognos
* Document Management (Virtual Case File) - Documentum, FileNet
* Optical Character Recognition, Intelligent Character Recognition – Captiva
* Address Standardization/Validation – Pitney Bowes
1. Under what circumstances does it make the best sense for a state child support agency to consider this new approach 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?

State Child Support Agencies with the following characteristics are best positioned to pursue the Platform as a Service (PaaS) approach for system modernization:

* Business requirements are well known, properly documented, and can be communicated to those responsible for configuring and customizing the platform to satisfy business needs. If business requirements are embedded in legacy code and not readily known by current staff, code conversion may be a better approach. However, Accenture is working on a modernization approach to convert legacy code directly to cloud-based platforms. Such an approach will help harvest business rules from legacy code, and will help reduce modernization timeframes and costs.
* Flexibility around how child support functions are handled, and processes are executed. Platforms are established frameworks and have inherent ways they provide specific features and achieve certain functions. It is advantageous to work within the constructs of the platform, leveraging out of the box features and relying upon configuration before customizing the platform. If the Agency is wedded to how its legacy system manages child support, the platform modernization approach may prove difficult.
* Willingness to outsource infrastructure. PaaS solutions are cloud based, therefore, pursuing a cloud approach is appropriate for agencies who want to get out of the hardware business, leverage an external party (provider) to manage and maintain the infrastructure, and eliminate their footprint in State Data Centers.
1. Generally speaking, what should a state expect on the following: project timeframe, project cost, time to rollout statewide?

**Delivery Timeframe:** This timeframe depends heavily on the platform approach taken (i.e. Platform integration vs. All-In Platform), and how much customization is required to satisfy business needs. An “All-In Platform” approach with standard customization will take a minimum of 3 years. The “Platform Integration” approach may reduce timeframes (i.e. due to reduction or elimination of platform customization), but other considerations arise like decreased performance, increased complexity, and reduced maintainability. If legacy code can be converted all the way to the PaaS solution, timeframes can be reduced without negative impacts on performance, complexity, and maintainability.

**Project Cost:** An “All-In Platform” approach with standard customization will be a minimum of ~ $50M. Increased customizations will drive costs up. “Platform Integration” may reduce costs (i.e. less platform customization), but same considerations noted above apply. Further, any savings realized from the integration approach may be offset by increased costs to support both the Platform plus integrated components. A legacy code conversion straight to PaaS approach (i.e. no integration for core components) would be the lowest cost solution. States may also realize savings if they transfer a PaaS solution from another State (when available), but savings are heavily dependent on flexibility to use the solution as is.

**Implementation Timeframe:** We recommend statewide rollout with the initial release, preceded by production simulation testing in targeted offices, counties, or regions depending on how the State is organized. Staggered rollouts require data synchronization between the legacy and new system, which can add significant time and cost to the overall delivery.

1. The states don’t want to again face the major system build and cost challenges once they have modernized. If they choose this approach, what is the continuous improvement model for the platform? Will the states benefit from the vendor efforts without major costs?

Platform as a Service (PaaS) solutions have extremely large user bases which drive ongoing investment and innovation. States with solutions on these Platforms benefit from the continuous innovation and automatic updates without additional cost. For example, Salesforce is a multi-tenant solution which introduces new innovative capabilities across three releases each year.  These enhancements are available to all customers, but those with solutions using fewer platform customizations are best positioned to take full advantage of these ongoing enhancements with little to no rework.

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

In preparation for this system modernization effort, States should:

* Identify business experts across all CSE core capabilities who can be business analysts during the system modernization effort.
* Clean up any known data issues in the legacy system in preparation for data conversion to the new system.
* Correct, clarify, or create legacy system documentation to help support the system modernization effort.
* Clearly define business requirements.
* Halt ongoing updates to the legacy system.
* Define test cases that, when run successfully on the new system, will build confidence across the user community.
* Define key stakeholders and organize a governing body who can efficiently review project issues and make decisions to help sustain forward progress during the modernization effort.
* Decide to pursue a “Platform Integration” (i.e. Platform integrated with other system components to provide core CSE capabilities) or an “All-In Platform” (i.e. all core CSE capabilities configured/customized directly in the platform) approach.
* Be flexible and ready to adapt your “how” (i.e. how you achieve the outcomes of child support processes) and possibly your “what” (i.e. your business requirements) to the capabilities of the chosen platform.
1. How does this type of child support system fit with states who need to have an enterprise approach? Many of the platforms seem to be creating the same old silos on a new platform. Is it possible to have one casefile for each person/family across the systems (child support, SNAP, TANF, family services, etc.)?

Child support systems already match common case files across IVD, IVA, IVE, Interstate, and Non-IVD intake processes.  Expanding this capability to include SNAP, IE, Child Welfare, and other programs is technically feasible, but requires large-scale organizational change and process reorganization across the human service agency (or agencies).  Without this strong governance across programs and agencies, the same siloed systems will be built regardless of the technology platform.

Once the organizational alignment occurs, cloud-based platform solutions will scale effectively to handle the increased case load and transaction volumes associated with an enterprise approach supporting additional agencies.  Further, States may benefit from transaction-based pricing models used with PaaS solutions, paying for increased capacity only when it is needed.  Cloud solutions scale capacity up and down. States will not need to carry the overhead costs associated with a large “on premise” infrastructure when the additional capacity is not needed.  State can realize lower cost and more efficient operations.  Citizens experience higher satisfaction because they no longer have to navigate the various programs and are truly at the center of their own Human Services experience.

1. What haven’t we asked that we should have?

What are the different ways to architect a Platform solution and what are the key considerations of each approach?

What are the disadvantages and key concerns with over customizing a Platform solution to satisfy CSE System requirements?

Child Support solutions have many external interfaces and batch processes, how will these be handled in a Platform solution?