

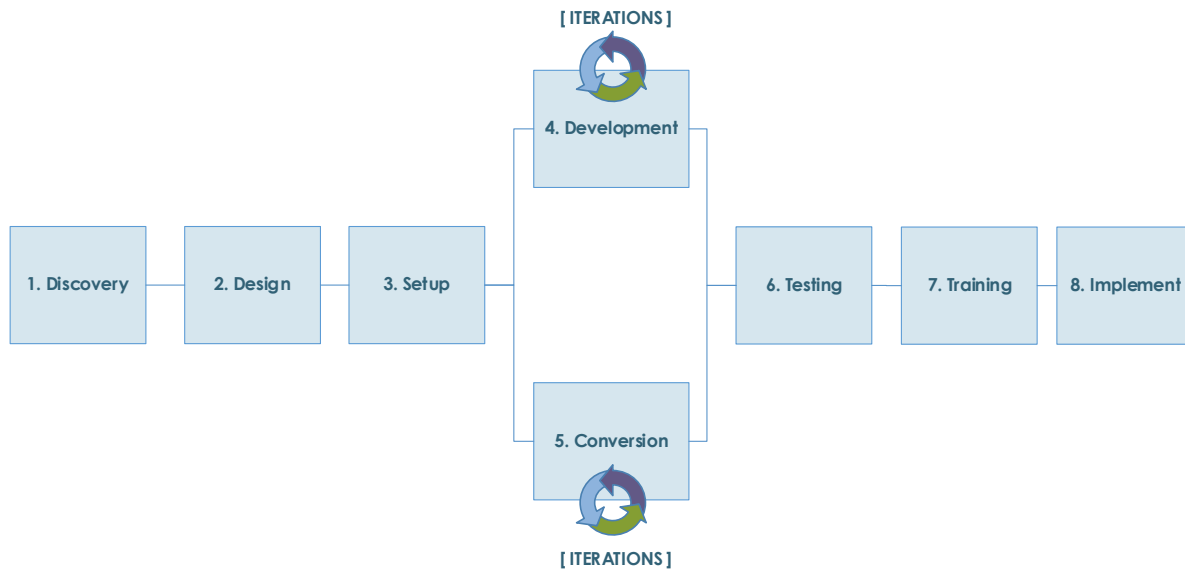
## Overview

**i** Our Proposal will walk through the business and technical details in a clear and concise manner using the structure prescribed in the RFP by the agency. We will explain our vision through this document and provide details on how we will be able to get there. This section and the ones to follow will include a mix functional, technical, and operational details.

With this proposal and the sections to follow, we have detailed all aspects of our proposed solution and its implementation. In the sections to follow, we have described every detail of what it takes to implement the functional and non-functional requirements within a systems integration project.

## Implementation Methodology

**i** Our implementation methodology combines the best practices of waterfall and agile techniques to deliver the proposed solution and associated deliverables. This is a proven methodology that we have used on multiple projects within the public sector to success. Our methodology provides a road map for successful implementation. It also facilitates work by providing a common language, leveraging existing examples of work products and artifacts, and building on a repository of best practices and lessons learned.



**Figure 5.3: Our proposed implementation methodology for System Integration Projects**

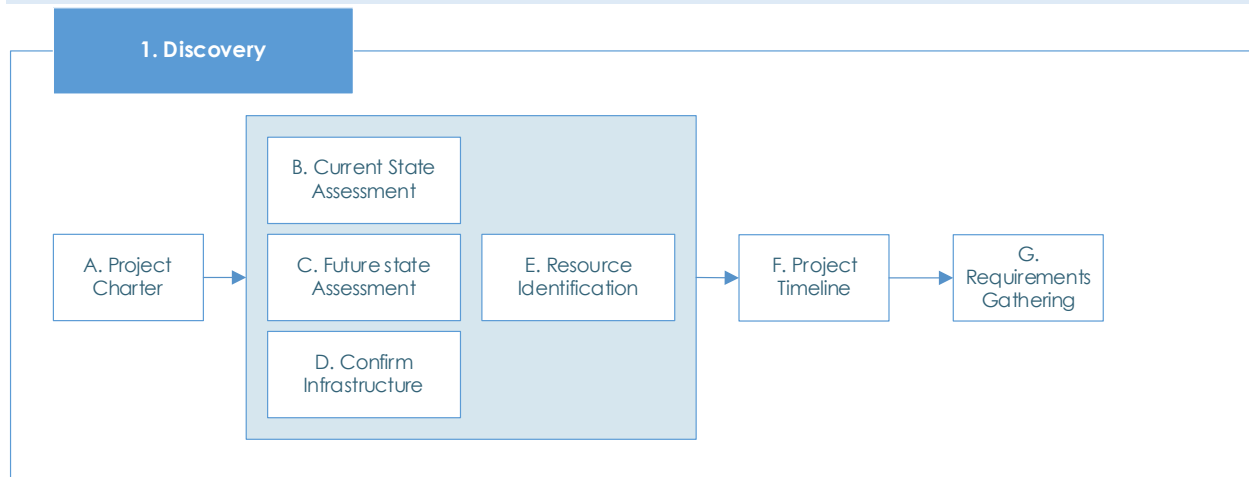
A fundamental principle in the methodology is the use of iterative cycles to complete certain activities. The underlying philosophy is to perform small increments of work, submit those for review and verification, incorporate the feedback from the reviews back into the next small increment of work and continue in this manner until the work products of a systems integration project are deemed acceptable. This approach ensures that incremental adjustments or changes required to stay on track can be made without retracing too many steps. It is important to note that we will execute some phases in parallel in our methodology. In some cases, we will start phases sequentially but end at the same time. Our team will

begin the project through an official kick off where we will go through the implementation methodology and how we plan to approach the project and its deliverables. Our methodology includes the following phases:

## Discovery

**i** This is the first phase of the implementation methodology. The objective here is to get a clear understanding of the project scope & detailed project deliverables. The outputs from this phase include:

- Project Charter
- RACI Chart
- Project Schedule
- Functional/Non-Functional Requirements



### *Project Charter*

The project charter is formal definition of scope. We will define a charter in the initial week through discussions with important stakeholders. The charter will cover objectives from the project, how we plan to accomplish them, who the stakeholders are on the project, any initial risks that have been identified for the project and proposed benefits from the project.

### *Current State Assessment*

As part of current state assessment, we will begin discussions with the current users to understand the pain points from the legacy system and the current operations. Our goal will be to understand how information flows and which stakeholders are involved at different points in the process. We will also review the current application from a business and technical standpoint. Our team will go through the documentation from the current implementation and the requirements that have been compiled so far for the proposed system.

---

### *Future State Assessment*

Future state is where the agency would like to go with the proposed project implementation. Our goal here is to define the future state from a business and technical perspective. We will document the findings in the form of functional and non-functional requirements from the project implementation.

---

### *Gap Analysis*

Based on the current state assessment and understanding the expected future state from the project, our team will complete a gap analysis using a combination of SWOT (Strength Weakness Opportunity and Threat) and Cause & Effect techniques. The result from this analysis will be an updated functional and non-functional requirement document for the proposed project. We will use this document to guide future activities.

---

### *Confirm Infrastructure*

Based on our understanding of technical requirements, we will be able to define the infrastructure needed to support our proposed solution. We will meet with stakeholders from IT Teams to understand the current infrastructure for the project. Our goal will be to know what infrastructure resources (location, network, hardware, development software, and associated logistics) are available for use by our team and agree on the deployment perspective i.e., on-premises, cloud, hybrid. We will also confirm, in case of cloud deployment, if the solution will be deployed using Infrastructure as a Service (IaaS) or Platform as a Service (PaaS) as the model.

---

### *Resource Identification*

For a project of this size and complexity, we want to partner with the agency and therefore understanding the resources that are available from the agency is very important. Resource identification will focus on knowing the resources that will be responsible, accountable, consulted and informed on important aspects of the project as we move through the implementation. We will organize the agency resources into focused product centric groups i.e., group responsible for module 1, group responsible for module 2, etc.

---

### *Project Timeline*

At this point in our implementation methodology we will have the necessary information to define a project timeline through all phases. The timeline will include project phases and the sub systems to be delivered with each phase. We envision having individual detailed project schedules as GANTT charts for each phase that tie back into the project timelines. We will monitor and manage the timeline through the project execution.

---

### *Requirements Gathering*

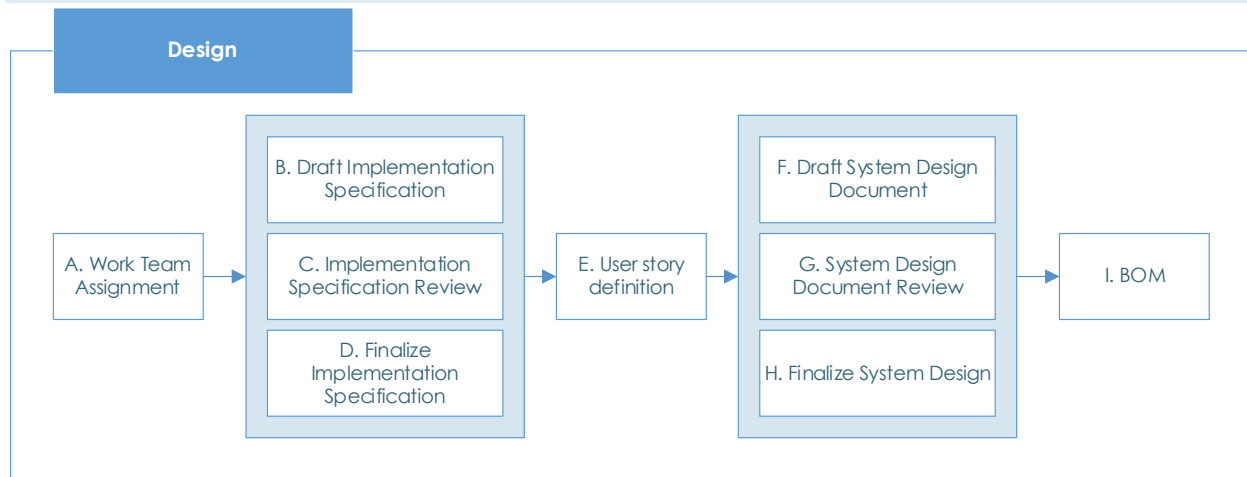
The final activity within Discovery will also be the longest. This will include collecting detailed requirements for the project. Our approach to collecting requirements include interviews and questionnaires with key stakeholders. At this point, our team has the functional and non-functional

requirements. We will use a template that allows use to understand these requirements in greater details, the actors involved with each requirement and the sequence of activities for the proposed project.

## Design

**i** *Application Design is often referred to as creating a blueprint for the proposed solution. As part of application design, we will draw out the functional requirements into implementation guidelines and the technical design elements that will support these guidelines. The outputs from this phase include:*

- Implementation Specifications
- User Stories
- Business Architecture Diagram
- Detailed Data Flow/Cross Functional Flowchart
- Technical Solution Architecture Diagram
- Application Architecture Diagram
- Database Architecture Diagram
- Security Architecture Diagram
- Deployment Architecture Diagram
- Operations Architecture Diagram
- Technical Design Specifications
- Project Bill of Materials (BOM)
- Low and High-Fidelity Wireframes
- UI/UX Style Guide



### *Work Team Assignment*

This activity is meant to organize our resources into teams assigned to specific functions with very clear responsibilities for the project i.e., Solution Architect, Team Leads, SMEs, Business Analysts, Designers,

Developers, Testers and Trainers. We envision have some resources spread across sub systems and some that are specifically assigned to a sub system.

---

### *Draft Implementation Specification*

An implementation specification is a formal design specification for business functionality. The implementation specification will include terminology, details on flow, actors, business rules, data elements, user experience expectations and the narrative details for functional requirements so that the reader/reviewer has one document to go through for a module or sub system.

---

### *Review Implementation Specification*

This is the process of reviewing the implementation specification with the agency users and SMEs. We will facilitate reviews using a walkthrough method where our team will go through the document using a series of meetings. At the end of each meeting, we will collect feedback and make necessary updates before reviewing the next logical section in the document.

---

### *Finalize Implementation Specification*

Once all the updates are in, we will provide the final implementation specification for approval by the agency stakeholders.

---

### *User Story Definition*

Implementation Specification is a document that will provide narrative details about a component and sub system. However, our goal is to engage the agency users through the development process in an iterative manner. User stories form the basis for iterative development cycle. We will use the implementation specification to create user stories. A single implementation specification will form multiple user stories. Every story will include details on the user involved, acceptance criteria to define the 'definition of done' We will associate the story back to the implementation specification so that there is traceability through project execution.

---

### *Draft System Design Document*

System design document is a technical document to capture the details related to non-functional requirements. Design document will include the business, technical, application, database, security, deployment, and operations architecture details.

---

### *System Design Document Review*

At this step, we will submit the design document for review. The process for capture updates will be like the Implementation Specifications Review sessions.

---

### *Finalize System Design*

Following review, we will make updates to the design document based on feedback. We will submit the final document for approval by the applicable stakeholders.

## BOM

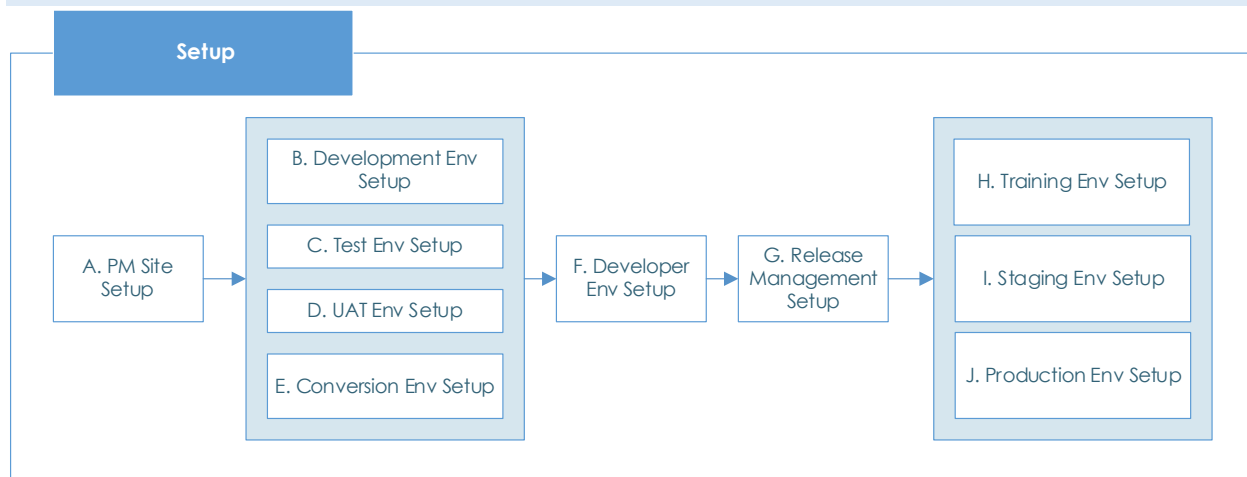
Following approval, we will compile the infrastructure resource requirements in a Bill of Materials (BOM) document. This will include:

- Hardware installation and configuration -
  - Physical server installation / hardening
  - Switch and routing
  - Virtualization
  - Firewall
  - Security
- Application Servers – OS install and load-balance configuration
- Database servers – OS and SQL Server installation and configuration
- Web service and file servers – file/folder security and web service configuration
- Disaster recovery planning and installation

## Setup

**i** Before development on a project can begin, we will work to put in place the necessary ecosystem to support development. This includes setting up the governance and communication platform, environments, and the coding/release management processes. The outputs from this phase include:

- Environment Specifications
- Code and Release Management Specifications
- Application Structure



---

### *PM Site Setup*

The Project Management Site is the cornerstone of our project governance and communications. This forms the repository for documents related to the project. It will also include project schedule in the form of milestones and tasks that must be completed to meet the milestones.

---

### *Development Env Setup*

This activity is related to setting up the development environment for our team to complete work on the components and sub systems.

---

### *Test Env Setup*

This activity is related to setting up the environment for testing the completed functionality by the QA teams on our side and possibly from the agency side.

---

### *UAT Env Setup*

This activity is related to setting up the user acceptance test environment where business users from the agency, providers and a focus group of parents can access the system.

---

### *Conversion Env Setup*

This activity will be applicable if data conversion is required on a project. With this we will setup the conversion environment which will be used for data migration related tasks.

---

### *Developer Env Setup*

On development projects, and our developers will require development software such as the Microsoft Integrated Development Environment (IDE) i.e., Visual Studio Team Services (VSTS). This activity will cover the code editing environment setup for developer staff.

---

### *Release Management Env Setup*

Our proposed solution will be developed using multiple release cycles. Each cycle will result in the completion of a specific functional module. We therefore need a mechanism to manage the build and release cycle along with the user stories for each cycle. This activity will cover the gated check-in, branching, build pipeline, release to different environments and notification setup.

---

### *Training Env Setup*

We will have an environment where our team can address the training requirements of the project. This activity will setup the training environment for the project.

---

### *Staging Env Setup*

Staging is the first of two higher environments and mirrors production. The goal of having the staging environment is to provide a location where performance testing, end to end testing and security testing can be completed prior to implementing the solution in production. The idea is to not have any surprises

when the solution is deployed for production use. We will setup the staging environment based on the BOM created in the prior phase of the implementation methodology.

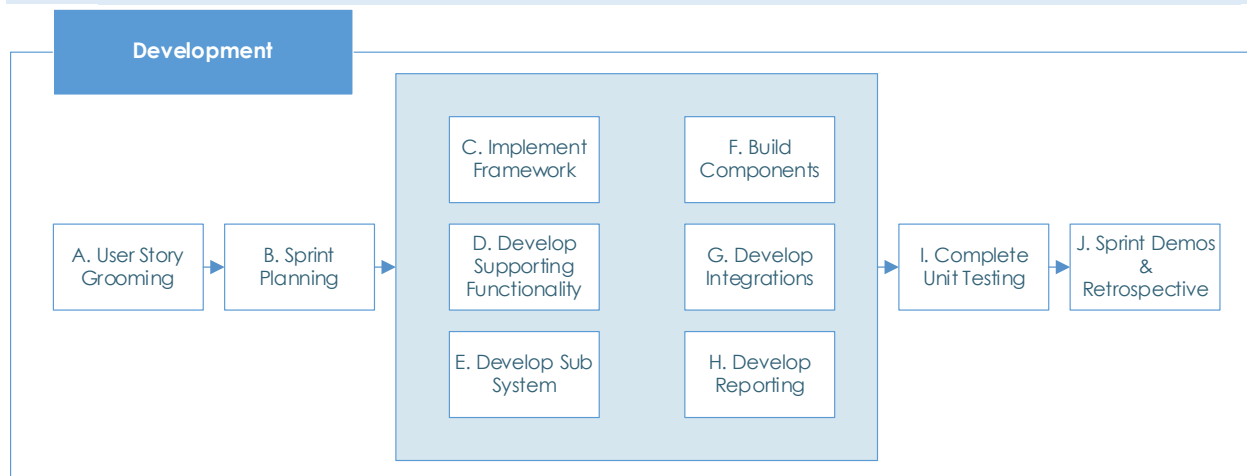
### *Production Env Setup*

Production is the second of the two higher environments. This will also be setup based on BOM completed during the Design phase. We will do the same level of configuration testing on the production environment as that completed on staging. Once verified, we will only deploy code to the production environment during the cutover weekend of the project phase.

## Development

**i** *Development phase is the process of actualizing the requirements into functionality. This will be the longest phase in our implementation methodology. It will also be the most important phase. We will approach development using multiple iterations. This will be highly recursive phase of the project where we will actively collaborate with the agency resources to ensure the functionality meets the project goals and we have not missed anything important from the project objectives. The outputs from this phase include:*

- Sprint Definitions
- Sub System Functionality Memos
- Component Functionality Memos
- Sprint Retrospective Memos



### *User Story Grooming*

Our implementation methodology allows changes during development process through the user story grooming activity. By this point, our team will have converted the implementation specification into user stories. Story grooming is the process of reviewing the stories constantly during development to ensure any deviations from the existing definition, are captured and the stories are updated to reflect that. We will therefore perform this activity as part of development.



---

### *Sprint Planning*

We will complete development using multiple iterations with each iteration focused on completing a specific unit of work or module. We refer to these iterations as sprints. Our goal with sprint planning is to identify the stories that will be included in each sprint and constantly assess performance. Based on this, we make resource adjustments to ensure we meet the project schedule.

---

### *Implement Framework*

As mentioned earlier, we will bring our common application framework as an accelerator for systems integration projects. This step will cover framework registration.

---

### *Develop Supporting Functionality*

The supporting functionality will cover the implementation of physical data model which will be developed in the following steps.

---

### *Develop Sub Systems*

Our goal here will be to build systems as modules. A sub system can include multiple modules within it. We will build these modules as part of this step.

---

### *Build Components*

At this step, we will package the individual sub systems into components.

---

### *Develop Integrations*

We understand that most development projects need to exchange information with other internal and external agency systems through integrations. We will develop these integrations using web services and file-based data exchange.

---

### *Develop Reporting*

Reporting and Analytics will be key component of our proposed solution. We will address the native reporting requirements from our solution as part of this step.

---

### *Complete Unit Testing*

We will follow a test-driven development approach. Every function that is developed will include a unit test case that is coded prior to coding the functionality. Our team will run the unit test on completing the functional module to ensure the functionality meets the specifications. Unit testing also helps with regression testing as we move through the development iterations.

---

### *Sprint Demo & Retrospective*

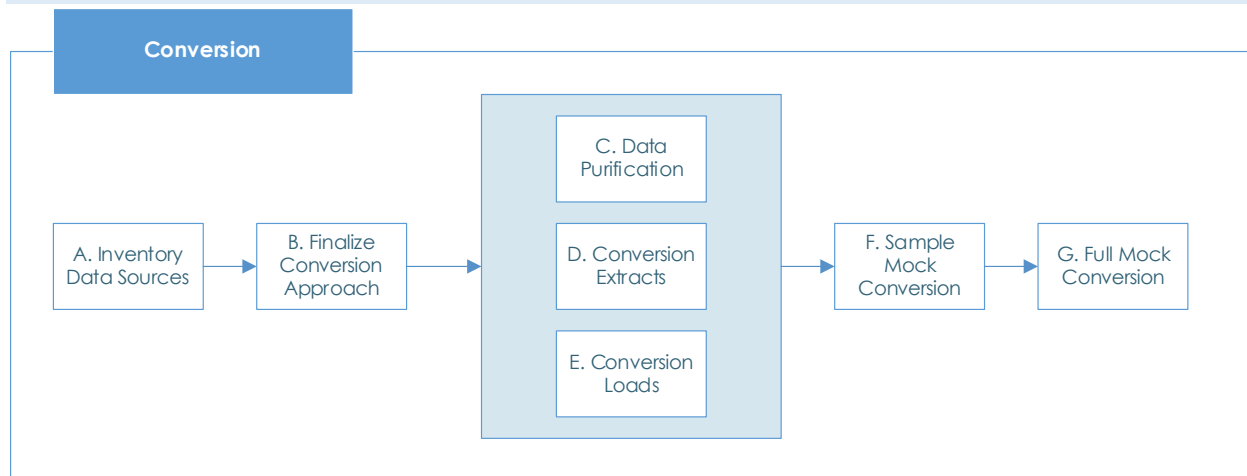
At the end of a sprint, we will demonstrate the functionality within the completed module. We will walk the user through the completed functionality and answer any questions during the demo. We will document any actionable feedback as modifications to the user stories to address in subsequent sprints.

Finally, following the sprint demo, we will convene a retrospective with the agency stakeholder to review the sprint and make necessary improvements going forward.

## Conversion

**i** This is an optional step within Application Development Lifecycle, and it depends on if the project requires data to be converted from the current or legacy system into the proposed data model. Conversion is an iterative process to extract, convert, purify, verify, and reconcile legacy data for use in our proposed solution. We will perform conversions using a series of mock conversions. In the Conversion Phase, legacy data from various sources is converted into the target database. We use the term conversion and migration interchangeably. The outputs from this phase include:

- Conversion/Migration Approach
- Data Purification Results
- Data Conversion Results



### Inventory Data Sources

The first step in data conversion is understanding all the data sources that feed the current system. For each of the components that are being converted, we will inventory the existing data resources. This will define the scope of agency data that is available to the conversion process. The data sources and our scope of conversion can include converting data as well as migrating documents. We will then review the data source for integrity and quality.

### Finalize Conversion Approach

The conversion approach artifact is our conversion team's equivalent to Implementation Specifications. It defines how the business data will be converted using the following decision points:

- Do we use manual versus automated processes?

- What is the statute of limitation on data i.e., do we need to convert data is older than 7 years?
- How much audit or historical data do we convert?
- Do we convert detailed or summary information?
- How do we handle work in progress or in-flight data?
- What is the impact on new system processes because of the different in workflow between current and our proposed solution?

---

### *Data Purification*

It is always better to make corrections to data that is part of conversion scope, within the legacy system. Data purification is the process of purifying or correcting information in the source system. Prior to the execution of the conversion, we will build processes to inspect the data from each source to determine any inconsistencies. We will assist in resolving these inconsistencies to purify the data prior to conversion.

---

### *Conversion Extracts*

We will build conversion extract processes using conversion scripts available through technologies like SQL Server Integration Services (SSIS). These scripts will draw data that is part of conversion scope and hold it in a secure temporary staging location. Our conversion extract process will also provide control reports that detail the extraction process and confirm the load processes.

---

### *Conversion Loads*

Once we have the extracts, we will implement data processing routines to load this information into our proposed solution data model. The conversion load process will validate the extract files to ensure the data format meets the agreed to structure. Following validation, our processes will transform the data to meet the new structure. Finally, the data will be loaded to the proposed data model in the new application, and our data conversion routines will produce conversion load control reports detailing the validation and load processes. Once converted data is loaded, we will convene data validation sessions where user can access the converted information from the completed functional modules of the proposed application.

---

### *Sample Mock Conversion*

Conversion for us is not a one-time event with this project. For a large project we will use multiple mock conversions. Mock conversions will be performed initially on a small subset of data and then on increasingly larger subsets until we have accomplished the conversion scope.

---

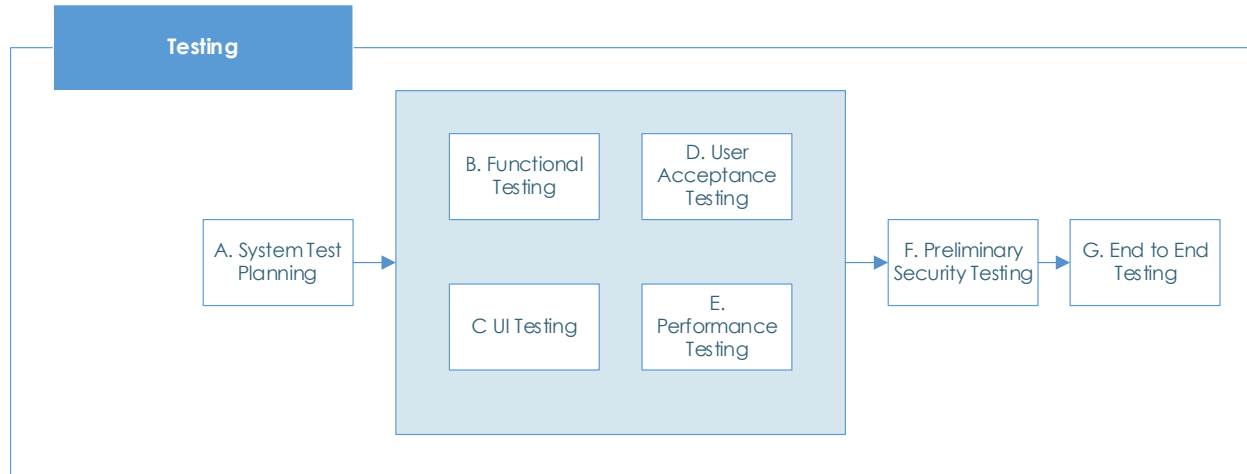
### *Full Mock Conversion*

We will perform full mock conversions in the days leading up to production deployment. One or more complete mock conversions will be done to provide the various stakeholders with an opportunity to test operational functionality with converted data. This is to ensure that the user can review the converted data and ensure the applicable downstream actions can be performed. Our conversion processes will provide a reconciliation document artifact that describes how converted data is reconciled to its legacy source.

## Testing

**i** Our approach to testing covers the formal testing requirements for proposed systems integration projects. With formal testing, we will cover functional and non-functional testing. Functional testing will cover application functionality. Non-functional testing will cover the performance and security testing. The outputs from this phase include:

- System Test Plan
- System Test Cases and Results
- Performance Test Metrics and Results
- Application Security Profile
- App Scan Results and Remediations
- End to End Test Scenarios and Results



### *System Test Planning*

Our System Test Planning will begin toward the end of the Development Phase, while unit testing is taking place. System testing is focused on testing business functions as opposed to individual screens or background processes. The System Test Plan identifies:

- Business test conditions
- Business test cycles
- System testing execution approach
- Modules targeted for performance testing
- End-to-end testing approach
- Acceptance criteria

### *Functional Testing*

At this step, our team will work with the agency QA resources to test the functionality delivered with all applicable sprints. To facilitate this process, we will develop functional test cases using the acceptance

criteria specified within the user story. Following the completion of testing, we will compile the results from the testing and make necessary updates to the relevant modules.

---

### *UI Testing*

At this step we will focus on multiple aspects of the user interface for our proposed solution. We begin this step by testing browser compatibility to ensure all solution components work consistently across all modern browsers. Following browser compatibility, we will test the application on multiple devices to ensure the solution UI adapts as expected on smaller devices. Finally, we will test all components of our solution for ADA compliance. The ADA compliance testing will review the HTML to ensure the HTML include tags that can be accessed by machine readers.

---

### *User Acceptance Testing*

Once functional & UI testing is complete, we will convene multiple user acceptance test sessions. We will have role-based walkthrough materials developed for these sessions. Our team will moderate these sessions and assist the users in reviewing functionality using the walkthrough documentation and answer any questions resulting from the test. If we see any valid defects or update requests, we will log it as a work item and review with the agency stakeholders. If approved, we will include these work items as part of an update to the solution before performance testing.

---

### *Performance Testing*

Our goal is to provide a solution that is robust and high performing. With each phase, we will complete a series of performance checks using a formal performance test or load test process. As part of this, our technical team will make a list of functions and queries that are intensive in priority order and align metrics for performance based on our contracted agreement with the agency. We will run these functions through a stress test using an automated tool. Based on the results from the test, we will make updates or optimizations to the solution. The performance testing scope will include:

- Intensive background processes
- Response times on all public facing functionality
- Response times on all resource intensive calls
- Response times for integrations
- Validation of network throughput during peak and off-peak hours

---

### *Preliminary Security Testing*

Just like performance testing, we also want to ensure our solution meets the data privacy & security requirements from the project. We will therefore complete a preliminary security test process. As part of security testing, we will create a profile for the application using a security assessment template. We will submit the details for review and convene discussions to ensure the security structure of our solution is reviewed. Following this, we will work with the agency technical resources to complete an application scan and a manual penetration test on our proposed solution. Based on the results of the scan and test, we will make updates to our application.

### End to End Testing

End to End Testing is the final step in our Testing Phase. End to End Testing will be completed in the staging environment with the agency's converted data from full mock conversions. This testing activity will use the full functionality of our proposed solution to replicate daily business activities with converted data in a Pilot Test environment. Our goal with this test will include:

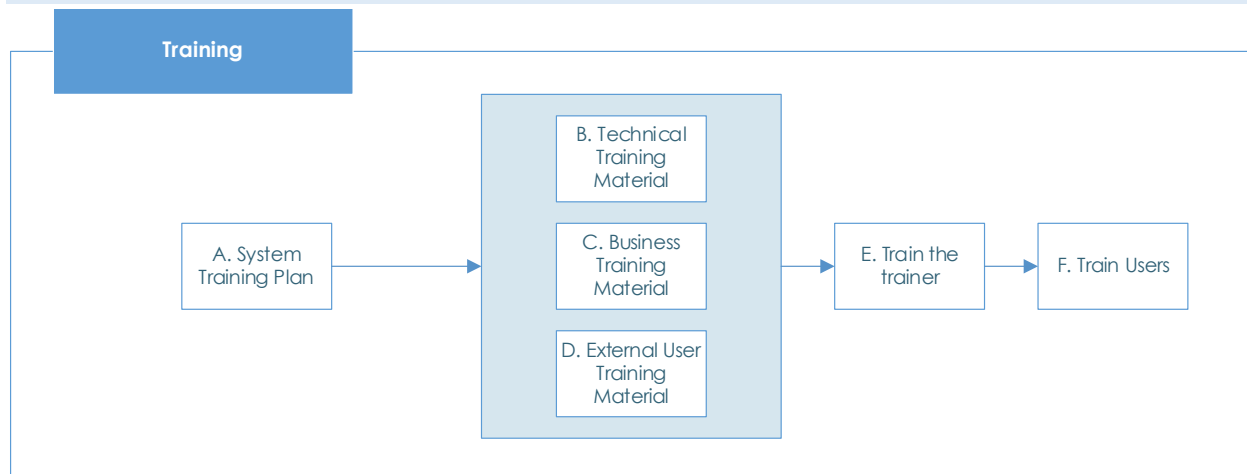
- Ensuring that the system correctly executes all business functions.
- Ensuring that the system correctly executes all external interfaces (inputs and outputs).
- Ensuring that scheduled jobs correctly execute required functionality.
- Ensuring that database adds, reversals, modifications, and tracing are correct.

During end-to-end testing, we will emphasize testing interfaces. Our team will be responsible to ensure all data interfaces work and exchange information exactly as specified in the implementation specifications.

### Training

**i** During the User Training Phase, business users will be given exercises that allow them to become acquainted with the new or modified business processes. This phase also includes the creation of detailed documentation of the new or changed processes. The new training material will include step-by-step instructions with thorough descriptions of associated system functionality and navigation instructions. Screenshots and links to other useful information will also be included in the training documentation. The outputs from this phase include:

- Detailed Training Plan
- Customized Product Technical/Operating Training Document
- Customized Product Functional Training Documents
- Customized Train the trainer document
- Product User Online Help Document



---

### *System Training Plan*

This is the first step with the training phase where our team will define a detailed training approach using a training plan document. We will review the document with the agency stakeholders. The plan will include approach for:

- Identifying trainers and trainees
- Selecting the training venue and equipment
- Deciding on the format - presentation versus hands-on
- Coordinating with testing activities
- Analyzing the impact on agency organization and scheduling
- Preparing the training schedule

---

### *Technical Training Material*

This activity will be focused on creating a series of documentation that will be geared towards a technical audience within the agency. The goal here is to ensure all information that is important in transitioning the system to in house resources is included. We will compile this information in close coordination with the technical team within the organization to ensure the details are understood.

---

### *Business Training Material*

This activity will be focused on creating user manuals for the end users. We will create training documentation as function specific lab exercises to ensure the users get a full picture of the functionality. Example of this can be "How to process an application for benefits?". Here we will illustrate in narrative details and through screen shots, the various steps, permutations involved in processing an application for benefits.

---

### *External User Training Material*

This activity will be geared to addressing the training requirements of users that we may not have access to. We will develop online learning documentation that can be accessed from the system. This will be contextual i.e., Help on how to reject an application, will be provided when the user is at that step in the application. In addition, we will also have user guides in the form of interactive video demonstrations of functionality within our system.

---

### *Train the Trainer*

This is a key aspect of our overall training approach. Once we identify key users within the agency that can be technical and business trainers, we will administer personalized training on technical and functional aspects of the system. Our goal here is to build enough know how within the agency on all aspects of our proposed solution.

---

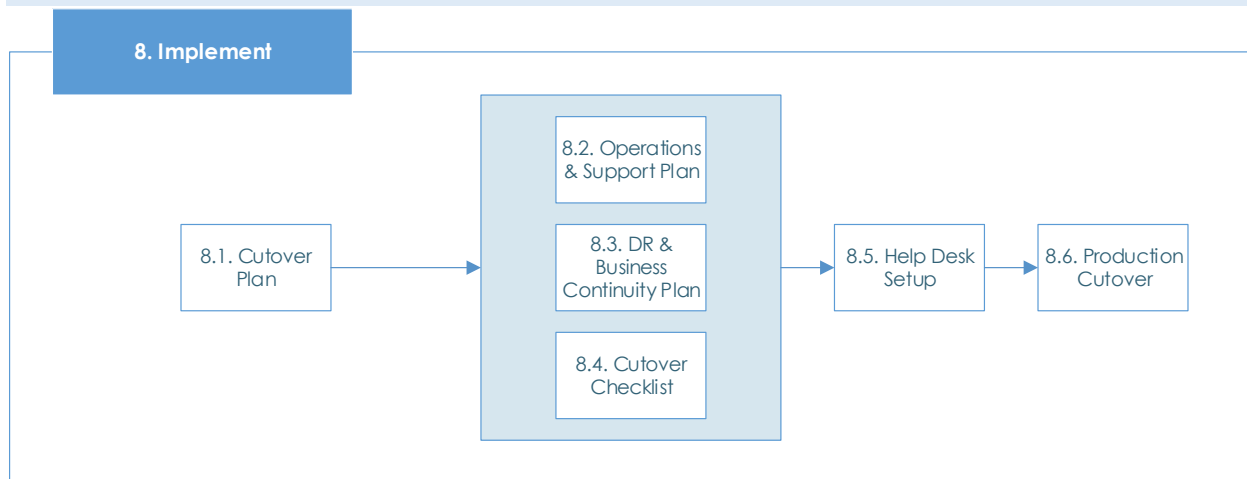
### *Train Users*

Here, our trainers and analysts will oversee a technical and functional training program to the support personnel and end users of our proposed solution. We will ensure that the agency trainers are present during the training sessions to learn from administering the training sessions.

## Implement

**i** This is also called the rollout phase. During this phase we will be focus on getting the final system ready for production. The outputs from this phase include:

- Operations & Support Plan
- DR & Business Continuity Plan
- Cutover Plan & Checklist



### *Cutover Plan*

The cutover plan is a detailed document showing the various tasks that must be completed for our solution to go live. The cutover planning meetings will be held every week in the final months of implementation to ensure everyone on the project is aware of the tasks assigned to different users and are held accountable for completing them.

### *Operations & Support Plan*

At this point, our team will have completed the training necessary for the technical teams within the client organization to understand the operations and support requirements from our proposed implementation. This step captures the same details in the form of document that is submitted to the project stakeholders for a final review and approval.

### *DR & Business Continuity Plan*

This is a key and often ignored aspect of software implementation. During the Discovery phase of our implementation methodology, we will capture the RPO (how much data can the agency afford to lose?) and RTO (how long can the application be down?) details during a disaster event for the project. Based on this information, we will submit a plan that covers the agency from a disaster and continuity of operations perspective. The plan will cover things such as replication, storage and compute requirements needed for our solution in a DR event.



---

### *Cutover Checklist*

This is condensed list of tasks that are focused on activities for the cutover weekend. This is obviously a very critical and often highly stressful event and will occur on a weekend. We will define a comprehensive step by step list of tasks with a responsible user for the tasks. The tasks must be executed in sequence and must be checked off by our administrators to ensure the cutover is a success. Examples of tasks on a checklist can include:

- Shut down the legacy system processes
- Shut down the applicable batch jobs
- Redirect the system interfaces
- Deploy the final solution
- Run a final full mock data conversion
- Send communications to agency management
- Complete smoke test on key transactions
- Revert status on smoke tested transactions
- Enable system notifications

---

### *Help Desk Setup*

Our Help Desk Setup will include a dedicated central support team and a desk-side support team comprised of project team members and agency experts in user work areas when the solution goes live. We will include the help desk contact details on the web application as well as publicize it as part of the cutover planning. The help desk will handle request from users and help in managing and resolving solution-related help calls. The users will also have access to desk side support to proactively resolve user questions. In cases where our help desk team cannot resolve a user question, they will dispatch a specialist from the project team to provide desk side assistance in person. As the number of solution-related calls to the help desk dwindle over time, our desk-side support service will be terminated, and we will handle calls centrally through the support period.

---

### *Production Cutover*

This is our go-live activity for each phase. We will run through the cutover checklist and this will result in our solution deployed to production and ready for production use. We will alert the help desk staff and deploy desk-side support to the different offices.

---

### *Production Support*

This is the post implementation phase of our project. Following implementation, the central help desk and the desk side assistance team will ensure the user questions are addressed promptly. Our experience has shown that deploying a desk-side support team is a highly effective means of managing anxiety and boosting confidence of agency staff as they first start to use the new system. During this time, we will work with the project stakeholders to understand any enhancement requests coming from the business users. We will document these are work items and go through a prioritization. Based on priority, we will deploy regular updates to the solution to include enhanced functionality or to resolve any identified defects.