

Section 4 Bidder's Products, Methodology, and Approach to the Project	4.1 FACTS II Requirements Summary	4.11 Interfaces
	4.2 Functional Requirements	4.12 System Development
	4.3 Technical Requirements	4.13 System Testing
	4.4 Customer Relations Management Tools	4.14 System Training
	4.5 Project Initiation and Management	4.15 Conversion
	4.6 System Hardware	4.16 System Implementation
	4.7 System Planning and Analysis	4.17 Post Implementation Support
	4.8 Requirements Verification	4.18 Support Federal Review
	4.9 System Design	4.19 Security
	4.10 Reports	

DE_SACWIS-002I_4

4.12 System Development

System development is a critical phase of the Delaware FACTS II project. The Deloitte team has extensive experience in successfully developing SACWIS systems of similar scope and scale. Most recent examples include Alabama SACWIS and the DC FACES.NET project. The Deloitte team uses the best industry practices and a collection of established tools, methodologies and frameworks throughout the development cycle in order to meet the Detailed Design Document specifications as well as to comply with the State’s IT standards and guidelines.

Deloitte’s proposed system development approach is in accordance with the requirements defined for Delaware FACTS II. Large systems integration projects are comprised of many inter-dependent elements. Success for such projects requires tight integration between the people, process, and technology components. An effective project management methodology, detailed plans, well defined design documents and efficient usage of tools brings together these components into a cohesive plan to achieve the defined goals for Delaware FACTS II.

The following subsections describe how Deloitte’s solution addresses the RFP requirements for system development, including the development methodology and tools, risk assessment and mitigation strategies, and description of the development deliverables.



section HIGHLIGHTS

- Deloitte has 18 years of experience developing seven SACWIS systems nationally
- Best industry practices, proven methodologies and frameworks
- Project members with SACWIS experience and technical expertise relevant to FACTS II development platform.

RFP reference: 6.12 System Development, Page 51

After the Department has approved the Detailed Design Document, the Bidder's development team will construct the application from the Detailed Design Document. The Bidder will provide the Department with a detailed project plan of development activities at the function level and with a schedule of code releases. The Bidder's development team will provide regular status updates to ensure that schedules are being met and that any issues are addressed in a timely manner. Additionally, the Bidder will create an Application Standards document that will incorporate the State of Delaware's standards, as well as the naming conventions, coding standards, version control processes, etc., that will be used by the development team. The Bidder's proposal should address the proposed development methodology and tools in detail. The Bidder should describe the Bidder's corporate level of experience using the proposed methodology, as well as the anticipated project team members' level of experience in the proposed methodology. Bidders should also describe their risk assessment, mitigation, and management strategies for the System Development phase.

System Development Methodology

Choosing the right tools and methodologies is an important step that lays the foundation for a developing a system with high quality standards. With industry experience, in depth research by development centers, the Deloitte team is uniquely positioned to ensure that this foundation is solid. Our proposed Delaware FACTS II solution utilizes Deloitte's proprietary Delaware FACTS II Playbook methodology shown in the figure below.

The proposed project team members are carefully selected from Deloitte's extensive resource pool, based on their actual field experience using these tools and methodologies. *Section 5.0, Organization and Staffing* provides the detailed qualifications of the anticipated project staff and their experience using these tools and methodologies. The tools, methodology and the overall approach proposed for the system development were specifically chosen for the Delaware FACTS II project based on:

- A thorough analysis of the RFP requirements for the new system
- Deloitte's experience with the SACWIS business processes, state-of-the art architecture and technology, and the agency operations specific to DSCYF
- A comprehensive software selection process performed by teams who have actually used the software on projects

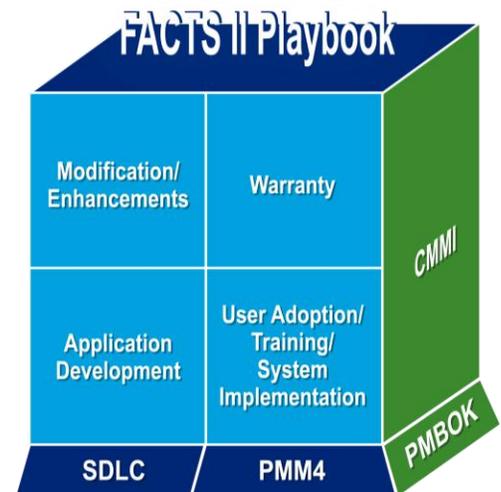


Figure 4.12-1. Deloitte's Delaware FACTS II Playbook Methodology.

This methodology is the vehicle for instituting procedures and controls on the Delaware FACTS II project. We have successfully applied this methodology to deliver hundreds of high quality projects. The Delaware FACTS II Playbook contains standard tools, detailed procedures, templates, standard work plans, status reports, and other materials that support all threads of the project including design, development, conversion, testing,

implementation and transition. By using the standards and templates contained in the Delaware FACTS II Playbook, we help the Delaware FACTS II leverage existing best practices, thus allowing the project team to focus their efforts on the unique conditions of this project. The use of this methodology reduces the overall technological risk and assist in the timely delivery of a project that is of the highest quality. Our system development methodology and approach provides the Delaware FACTS II project with the following benefits shown in the table below that were the foundation of FACES.NET and Alabama SACWIS projects. We work with the DSCYF to define components of our Playbook methodology to fit for Delaware FACTS II.

Feature	Benefits
Development Productivity Tool Suite	<ul style="list-style-type: none"> • Supports rapid application development using .NET application development frameworks
Structured Quality Assurance Processes	<ul style="list-style-type: none"> • Achieves compliance with code standards via templates, automated tools, and checklists • Peer reviews, QA checks, and developer forums communicate and emphasize software quality
Experience in SACWIS solutions nationwide	<ul style="list-style-type: none"> • Proven methodologies, tools and processes for faster implementation • Less risk to implementation due to prior experience and best practices used from other projects
Leverage DC FACES.NET architecture, system design artifacts, and framework to develop the Delaware FACTS II Solution	<ul style="list-style-type: none"> • Enables seamless transition and accelerates development • Delivers standardization and consistency in deliverables

Table 4.12-1. Features and Benefits of Our Development Approach.

System Development Tools

Deloitte has also developed a set of project management tools to meet a range of requirements, including long-term project planning, initial startup, and detailed daily project activities. The tool set consists of specialized in-house and commercially available applications developed by leading software vendors. From the beginning to the completion of the project, we leverage a variety of development tools that support our system development approach. We use a variety of automated tools to help us execute our system development approach.

SACWISmate

As mentioned in *Section 4.8.8* and *Section 4.9*, we propose to use SACWISmate, our project management tool designed specifically for SACWIS projects during the requirements and the design phase of Delaware FACTS II. This holds true for the System Development phase also. SACWISmate is the repository using which developers perform development of the application as per the requirement, the design and the associated business rules in housed in SACWISmate. SACWISmate facilitates the management of various activities during the system development phase. The tool is a comprehensive, integrated database and is built to support each SDLC phase (software

development life cycle) as well as integrate the tracking of information across phases. For example, SACWISmate documents system requirements, design documentation, test scripts, defects, enhancements and time spent on system modules. From these, summary and detail reports can be generated. The different components are linked together so that requirements are linked to designs, test scripts, issues, and incidents, so that they do not fall through the cracks and are fully traceable through the development life cycle. Additionally, large sections of many deliverables can be generated directly from SACWISmate, including the Detailed Design Document which is used for System Development.

The following figures show the Design module of SACWISmate.

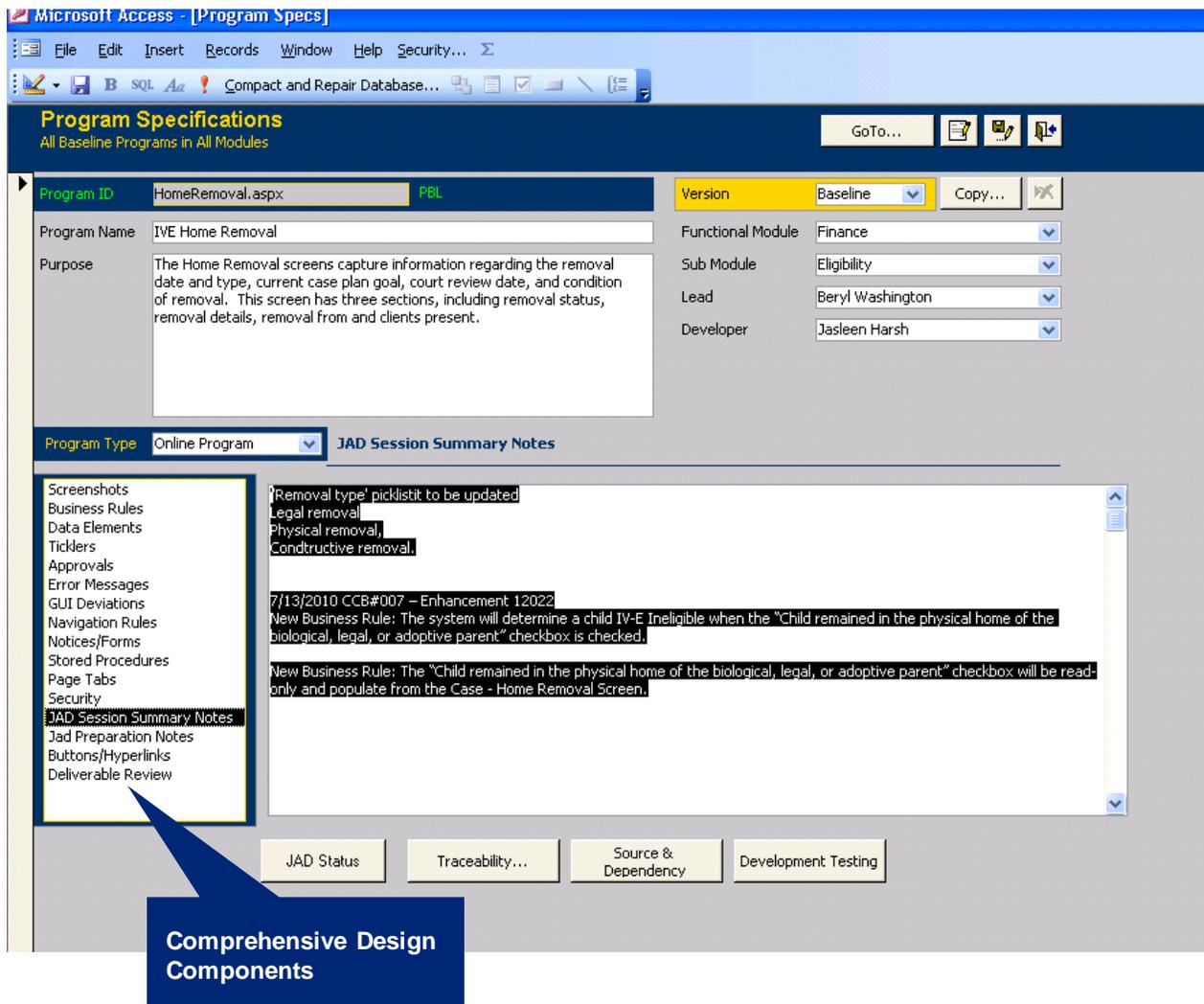
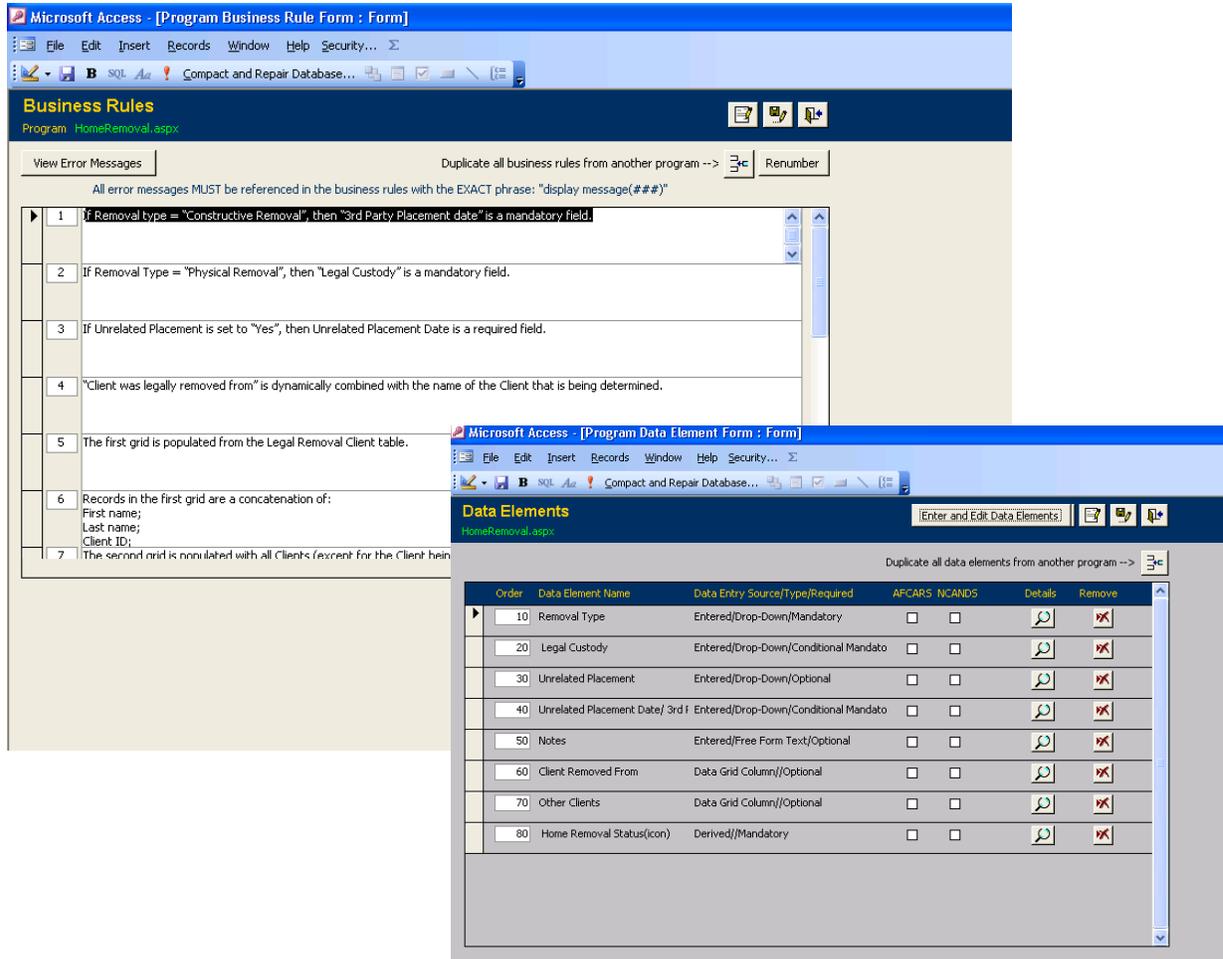


Figure 4.12-2. Design in SACWISmate.

DE_SACWIS-199



DE_SACWIS-198

Figure 4.12-3. Ease of Access to Design Information for Development Team.

The Detail Design module serves as a single repository for all Delaware FACTS II design documentation such as business rules and data element mapping that is easily accessible to the entire project team.

Integrated Application Development Environment (IADE)

We chose to develop DC FACES.NET solution in Microsoft's .NET framework, because it offers a more productive and cost-effective application platform for object-oriented development. We bring the same toolset for Delaware FACTS II that includes Visual Studio .NET with C# as programming language. The use of a .NET framework and Microsoft Visual Studio 2010 (VS 2010) cuts the time and cost of development and operations allowing users to take full advantage of robust applications within a development cycle.

The .NET technology offers large, scalable, high-availability application server platforms and practical frameworks for object-oriented component-based application development and deployment. Visual Studio 2010 .NET IDE provides easy to use, rich Graphical User Interface tools that facilitate design, development (code editing), compiling and debugging of applications from within a single window. It also supports object-oriented design and programming principles and provides an easier way to create and deploy web services.

Programmers have a choice of working in multiple programming languages such as C-Sharp (C#), VB.NET, F#, C++, Visual Basic, and it also supports other languages such as M, Python and Ruby through separate installable.

Deloitte has been using Visual Studio, Team Foundation Server on our DC FACES.NET solution as a source code editor, compiler, debugger, version control system and for automated build and deployment processes.

Source Code Control and Configuration Management

Managing the amount and variety of source code components and project deliverables for a project of the size and complexity of the Delaware FACTS II project requires the use of software tools. We propose to use Microsoft Team Foundation Server (TFS) which helps manage the tracking and versioning of software and other project deliverables.

Team Foundation Server. Team Foundation Server (TFS) is a Microsoft product offering source control, data collection, reporting, and project tracking, and is intended for collaborative software development projects. It is available either as stand-alone software or as the server side back end platform for Visual Studio Team System (VSTS). TFS is a configuration management tool that supports parallel development by providing locking, versioning, synchronization capabilities. TFS toolset integrates directly with our proposed integrated development environment, Visual Studio, so that the developer can perform configuration management functions without leaving their development tools. In addition, TFS provides traceability by recording actions that affect a project.

Our System Development Approach

The Deloitte team recognizes the positive impacts a robust and detailed system development approach can have on the long-term success of a project. Our software development approach has been proven to work on engagements across multiple Deloitte software implementation projects. Deloitte proposes a waterfall approach to development that supports the ability to trace the software changes made as part of defects and/or enhancements back to the design artifacts and requirements. Deloitte has successfully used this methodology extensively in other SACWIS and large-scale HHS system implementations. Our proposed development approach constructs Delaware FACTS II using a step-by-step approach in which discrete components containing logical groupings of functionality are developed, unit tested, and integration tested.

The following software development principles form the core of the Deloitte proposed development approach for Delaware FACTS II:

- **Develop Software in a Step-by-Step Manner.** Our customized System Development approach allows for the selection and deployment of selected business workflow, framework and application components needed to support identified business processes. The content of each step is driven by a prioritization of functional and technical requirements identified by DSCYF during the requirements and design phases.

- **Manage Requirements.** With our customized approach, all stakeholders gain a common understanding of the application components that are developed and tested. All requirements are managed using SACWISmate.
- **Use Component-Based Architectures.** Our methodology helps identify, isolate, design, develop, and test core application components upfront within the development life cycle. These components can be individually tested and integrated to form the Delaware FACTS II system. Some of these components are developed to be reusable components that provide services to many application modules within overall system.
- **Continuously Verify Software Quality.** Quality management, verification and validation is a core and ongoing activity, integrated into all phases of Deloitte team's development approach. Every member of the Application Development team understands their responsibility and contribution towards the overall quality of the application. Our collaborative approach working with DSCYF, coupled with our extensive experience in implementing SACWIS solutions allow us to maintain.
 - Product Quality by delivering a System that meets the DSCYF's needs
 - Process Quality which includes the quality of artifacts and deliverables like Test plans, etc, produced to support the development and implementation of the platform
- **Control Changes to Software.** During the development phase, software and supporting artifacts undergo frequent modifications. We use sound configuration management techniques and tools like Visual Source Safe to enforce industry best practices ideally suited for development. In addition SACWISmate facilitates tracing of requirements and demonstrates requirements coverage through design, development and implementation of FACTS II. The complexity of the development phase and all related activities can only be managed effectively if there is a consistent approach to each activity. We are confident that our approach provides the flexibility to establish an overarching application development process with activities that are carried out in a step-by-step manner to achieve the overall functionality for FACTS II.

RFP reference: 6.12 System Development, Page 51

With FACTS II, the Department desires a state-of-the-art systems development environment that complies with State of Delaware standards for Coding Languages, COTS components, etc. Any COTS components selected must also comply with the Federal 45 CFR 95.16(c) these standards will be provided at the Bidder's conference after receipt of a signed non-disclosure agreement.

After the Department has approved the Detailed Design Document, the Bidder's development team will construct the application from the Detailed Design Document. The Bidder will provide the Department with a detailed project plan of development activities at the function level and with a schedule of code releases. The Bidder's development team will provide regular status updates to ensure that schedules are being met and that any issues are addressed in a timely manner. Additionally, the Bidder will create an Application Standards document that will incorporate the State of Delaware's standards, as well as the naming conventions, coding standards, version control processes, etc., that will be used by the development team.

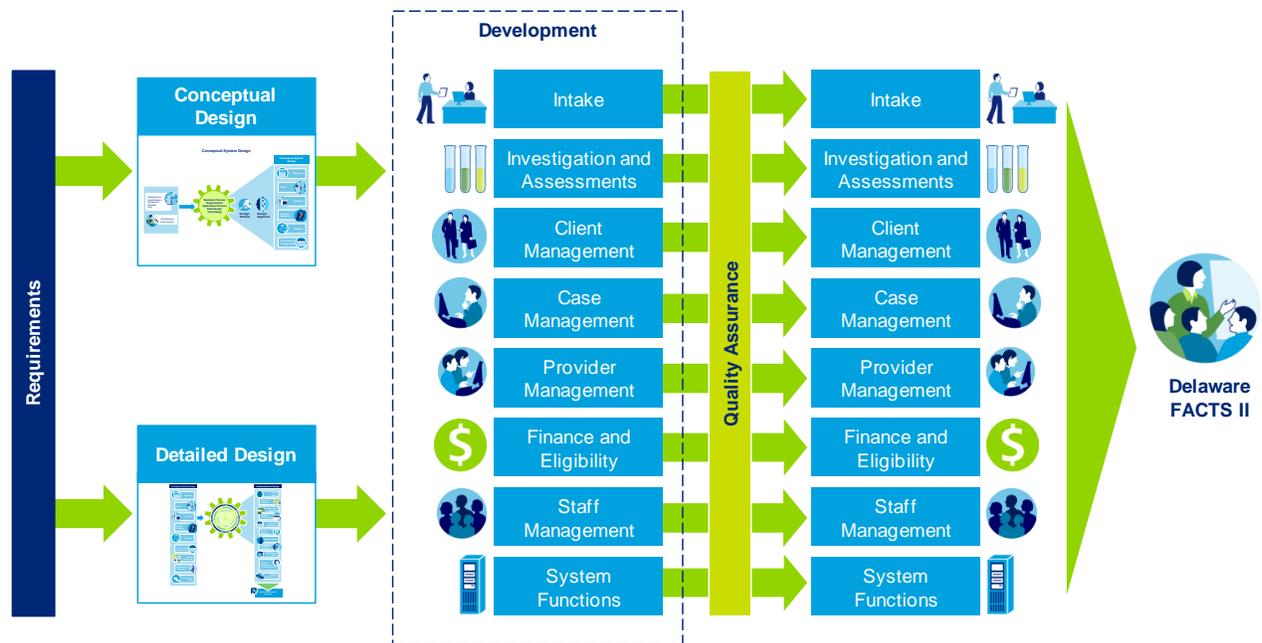
The Bidder's proposal should address the proposed development methodology and tools in detail. The Bidder should describe the Bidder's corporate level of experience using the proposed methodology, as well as the anticipated project team members' level of experience in the proposed methodology. Bidders should also describe their risk assessment, mitigation, and management strategies for the System Development phase.

Development Environment

For maintaining Delaware FACTS II operation, the development environment is the main component to promote changes and code fixes to maintain and expand the system functionality. The development environment is dedicated for developers to complete most of their work.

System Development Methods

Given the inherently comprehensive nature of a large-scale project, it is imperative that we provide structured methods to monitor critical development phase processes, including structured unit and integration testing, peer reviews, and Quality Assurance (QA) checks to mitigate risks toward an on-time delivery. The figure that follows depicts the transformation of development inputs to deliver Delaware FACTS II.



DE_SACWIS-201

Figure 4.12-4. The Development Phase of the Delaware FACTS II Solution.

During the development phase, requirements and design become the Delaware FACTS II solution through the execution of structured development processes that emphasize software quality.

The key development activities that result in the construction of Delaware FACTS II include:

Key Activities	Description
Development Preparation	The requirements gathered and conceptual and detailed design documents are reviewed and clearly understood before developing the required functionality. Requirements and design are mapped to the code modules and assigned to the responsible teams.

Key Activities	Description
Code	<p>The activity involves generating source and object code, executables, and scripts for the software development. Developers write the source code in accordance with the programming standards established in collaboration with DSCYF and according to the specifications documented in the detailed design document. Regardless of the platform, code development is to adhere to a consistent set of programming techniques and error prevention procedures. This promotes reliable and maintainable code, developed in the most efficient and cost-effective manner. Development staff also coordinates the following activities:</p> <ul style="list-style-type: none"> • Reference Tables • Database Changes • Reports • Batch Programs • Business rules
Unit Testing	<p>Unit testing is used to verify the input and output for each component. Successful testing indicates the validity of the function or sub-function performed by the component and confirms traceability to the design. During unit testing, the developer tests each component individually and verifies the component interfaces for consistency with the design specification. During unit testing, actual results of the important processing paths through the component are compared with expected results. In addition, the developer tests error handling paths. Adherence to standards is measured through documentation of unit test checklists.</p>
Quality Assurance Activities	<p>The testing team performs a quality check to determine compliance with the methods, standards and guidelines of all for newly created or modified code files that has been checked in by the developer.</p>
Manage Requirements and Update Design Documents	<p>Deloitte Functional Leads review the components developed and cross-reference them with those in the Requirements Traceability Matrix. Design is updated as appropriate as additional technical details are defined during the development phase.</p>

Table 4.12-2. Key Activities of the Development Phase.

Development Preparation

Development Preparation is the first activity which is carried out at initiation of the Development phase. This activity is extremely crucial in order to avoid unnecessary delays in the development phase and understand the design and requirements upfront. Deloitte functional leads work with the lead analysts and developers to map requirements and design to the code modules and understand the required functionality to be coded. The Deloitte team works in collaboration with DSCYF to review any concerns or questions at this stage before development can commence. Only on thorough review and approval by DSCYF of the design does the coding begin.

Code

In this activity the information gathered during the requirements and design phases is constructed to produce code components which integrate to deliver the overall Delaware FACTS II solution. The activity involves generating source and object code, executables, and scripts for the software modifications. Developers write the source code in accordance with the programming standards established in collaboration with DSCYF and according to the specifications documented in the detailed design document. Regardless of the platform, code development adheres to a consistent set of programming techniques and error prevention procedures. The development team is responsible for all development components including:

- Reference Tables
- Database Changes
- Reports
- Batch Programs
- Business rules

Unit Testing

Unit testing is a key exercise that is performed during the development phase before functionality can be promoted to the System Testing environment. The main objective of unit testing is to verify that the individual code component meets both business and technical design specifications. Development staff use the detailed design specifications to unit test each program or component developed and conduct unit testing based on the development work plan. Development staff uses the unit test checklist to adhere to standards.

Effective unit testing delivers more efficient, stable software to testing environments. Any discrepancies identified during the unit test are corrected with reassessment to confirm that the functionality conforms to the requirements. The development team documents issues discovered in unit test in SACWISMATE with problem statements, root cause, and resolutions providing visibility to DSCYF and Deloitte Functional Leads about the types of issues found in this stage of testing and valuable information for reference in later test phases should similar or related issues be identified.

Quality Assurance Activities during Development

During development the new or updated code modules are reviewed against Delaware FACTS II standards and guidelines to deliver quality code which is compliant with the coding standards. This process is practiced for all code files checked in to the development environment by the developers. This approach helps maintain code standards and reduces risk of changing the application following the quality assurance process.

Manage Requirements and Update Design Documents

The Deloitte team understands the importance of establishing broad documentation to support new development and enhancements within the project. This documentation provides a clear understanding and traceability of new or modified requirements and also avoids avenues of miscommunication that may pose a potential risk to project deadlines and stability of the system. We further recognize that the development phase especially calls for consistent updates to deliverables to meet the goal of achieving system development while maintaining knowledge about and traceability to validate requirements.

Another consideration in this approach is the decision of when a document alteration is necessary. This initial undertaking is set in motion by the modification of code due to new information that is a modification to the requirements. As these system enhancements or maintenance activities are implemented, key resources from Deloitte and DSCYF work to update the documents that correlate to the identified development item and effort level and confirm that they are submitted in a timely manner and to the appropriate parties.

While updates to existing formal documentation are important to DSCYF, the Deloitte team also sets this expectation at the development level for an additional layer of diligence. We proactively perform documentation activities within the source code regardless of the scale and effort of the development task to create a distinguishable trail of the modifications that occur. Through these various methods, we are confident that DSCYF is supplied with the most up-to-date, accurate items that support DSCYF business processes.

We recognize that documentation is important and in most cases, maintenance of documents requires manual intervention and effort. For this purpose, the Deloitte team uses SACWISMATE for generating and updating detail design documentation for the Delaware FACTS II project.

Application Standards Document

The Deloitte team recognizes the importance of coding standards and guidelines to maintain quality and uniformity in developed code as discussed above. Established code standards and guidelines improve the readability and maintainability of application code. Deloitte team collaborates with DTI to define standards and guidelines prior to the commencement of the development phase. Deloitte creates an Application Standards document deliverable that will incorporate the DSCYF standards, as well as the naming conventions, coding standards, version control processes, etc., that is used by the project team. These are enforced by using standard QA processes, and checklists. All COTS components as listed in *Section 4.3, Technical Requirements*, are also verified to comply with the Federal 45 CFR 95.16(c) guidelines.

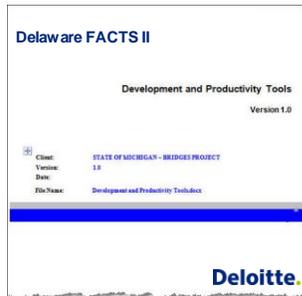


Table of Contents	
1.	INTRODUCTION 8
1.1	PURPOSE..... 8
1.2	SCOPE..... 8
1.3	DEFINITIONS, ACRONYMS, AND ABBREVIATIONS..... 8
1.4	OVERVIEW..... 8
2.	CODE ORGANIZATION AND STYLE..... 9
2.1	PARAGRAPH INDENT YOUR CODE..... 9
2.2	WSAD FORMAT CODE FACILITY..... 9
2.2.1	Settings in WSAD..... 10
2.3	BLANK SPACES..... 11
2.4	BLANK LINES..... 12
2.5	WRITE SHORT, SINGLE COMMAND LINES..... 12
2.6	LINE WRAPPING..... 12
2.6.1	Aligning the new line in case of an expression..... 12
2.6.2	Breaking an arithmetic expression..... 12
2.6.3	Wrapping method declarations..... 13
2.6.4	Wrapping lines and expressions..... 13
2.7	NUMBER OF LINES..... 13
2.7.1	Classes..... 13
2.7.2	Methods..... 13
2.8	STATEMENTS..... 13
2.8.1	Simple Statements..... 13
2.8.2	Compound Statements..... 14
2.8.3	if, if-else, if else-if else Statements..... 14
2.8.4	for Statements..... 14
2.8.5	while Statements..... 14
2.8.6	do-while Statements..... 15
2.8.7	switch Statements..... 15
2.8.8	try-catch Statements..... 15
2.9	IMPORTING CLASSES AND PACKAGES..... 16
3.	DECLARATION, PLACEMENT, AND INITIALIZATION..... 16
3.1	DECLARATION OF VARIABLES..... 16
3.2	USE PRIMITIVE VARIABLES..... 16
3.3	INITIALIZATION OF LOCAL VARIABLES..... 16
3.4	USE JAVA PRIMITIVE DATA TYPES..... 16
3.5	DO NOT CREATE UNNECESSARY STRING OBJECTS..... 16

DE_SACWIS-200

Figure 4.12-5. Application Standards and Guidelines.

Code standards and guidelines validate software quality per industry standards and improve maintainability of the application.

The Deloitte team establishes programming standards and guidelines and employs manual reviews as well as third-party tools to safeguard around non-compliances. Our Quality Assurance (QA) team conducts walkthroughs, inspects code, monitors and tracks quality, and prepares reports that are reviewed by senior development staff and project management on a routine basis.

The Deloitte team submits and reviews development standards and guidelines to be used for the construction of Delaware FACTS II solution. These guidelines use standardized checklists to verify that developed code adheres to requisite standards. Developers use these checklists for .NET, Batch, Reports and Interfaces programs development. Our QA team reviews these before approving code and promoting it to a higher environment.

Quality Assurance Activities

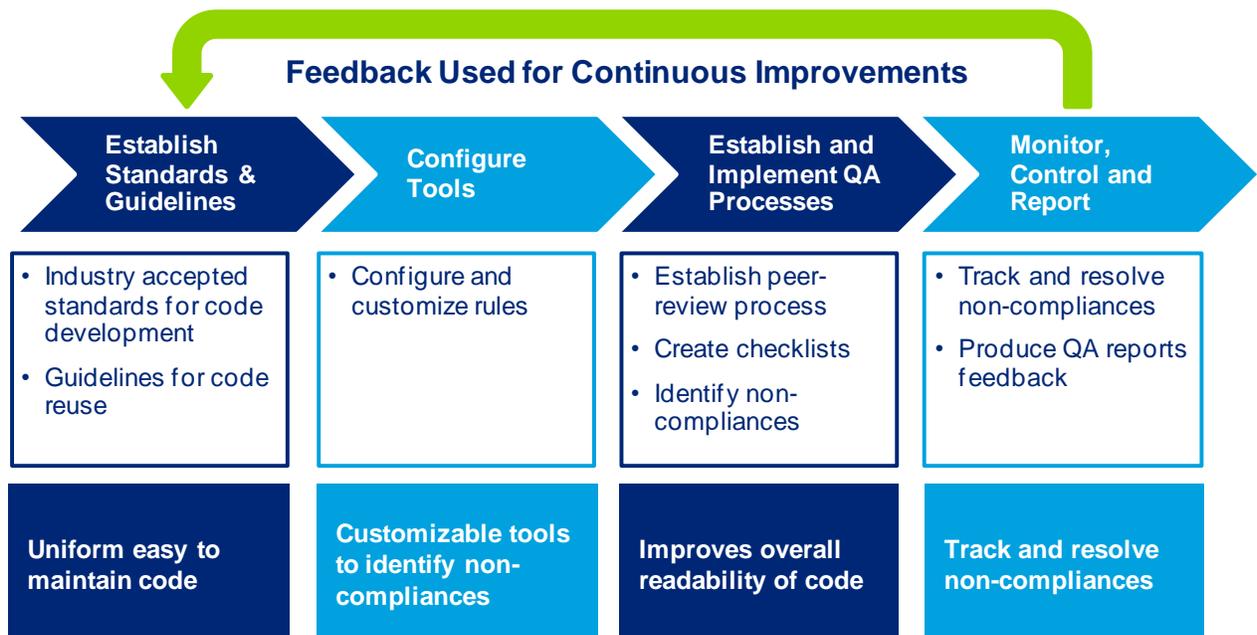
We use multiple techniques to enforce the coding standards and guidelines established mutually with DSCYF. These techniques include peer reviews, automated tools, and QA audits. We also create reports to document our compliance with coding standards and coverage across the code base.

We emphasize the use of manual reviews to confirm that in-line code documentation and code comments meet coding standards and guidelines set forth for the project. After the developer completes unit testing, development peers and members of the QA team review the code for consistency and accuracy. Peer reviewers and the QA team members

perform reviews with checklists to confirm the use of in-line and code comments. If the application code does not meet coding standards, the peer reviewer and the QA team documents the results of the review and works with the team lead and developer to resolve issues before the code changes are promoted to any other environments.

To enhance the knowledge and understanding of the entire development staff, Deloitte team conducts “Developer Forums/Development Team Meetings” to review coding standards and guidelines, address concerns identified during QA checks, and distribute information regarding changes in framework functionality.

The following figure illustrates our approach to enforcing software quality.

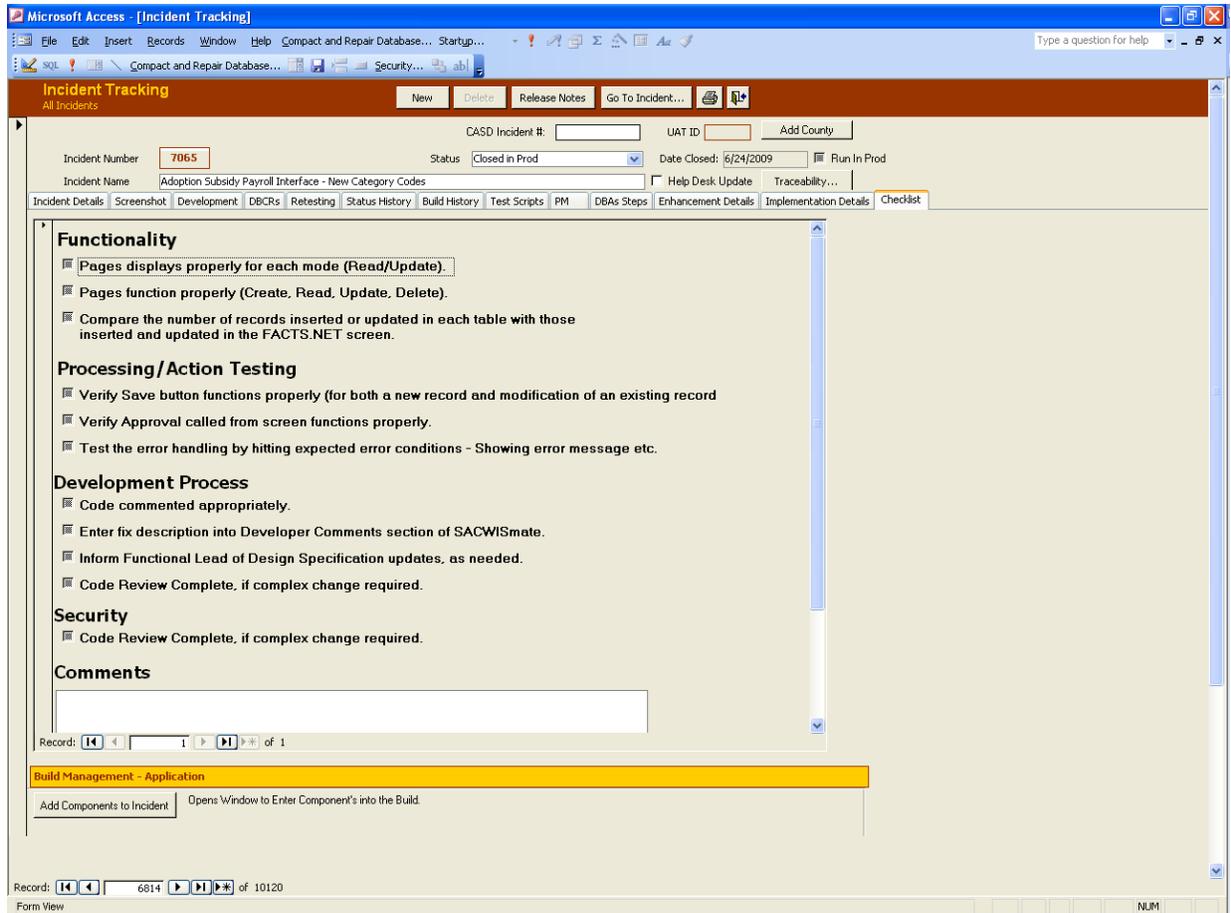


DE_SACWIS-197

Figure 4.12-6. Software Quality Approach.

Peer reviews, automated tools, and QA checks validate software quality in a release, continuously improving the overall quality.

As part of the unit testing process, development staff completes unit testing checklists to promote consistency and provide descriptive documentation of each change to the system. The figure below displays a sample of a unit test checklist.



DE_SACWIS-107

Figure 4.12-7. Unit Test Checklist.

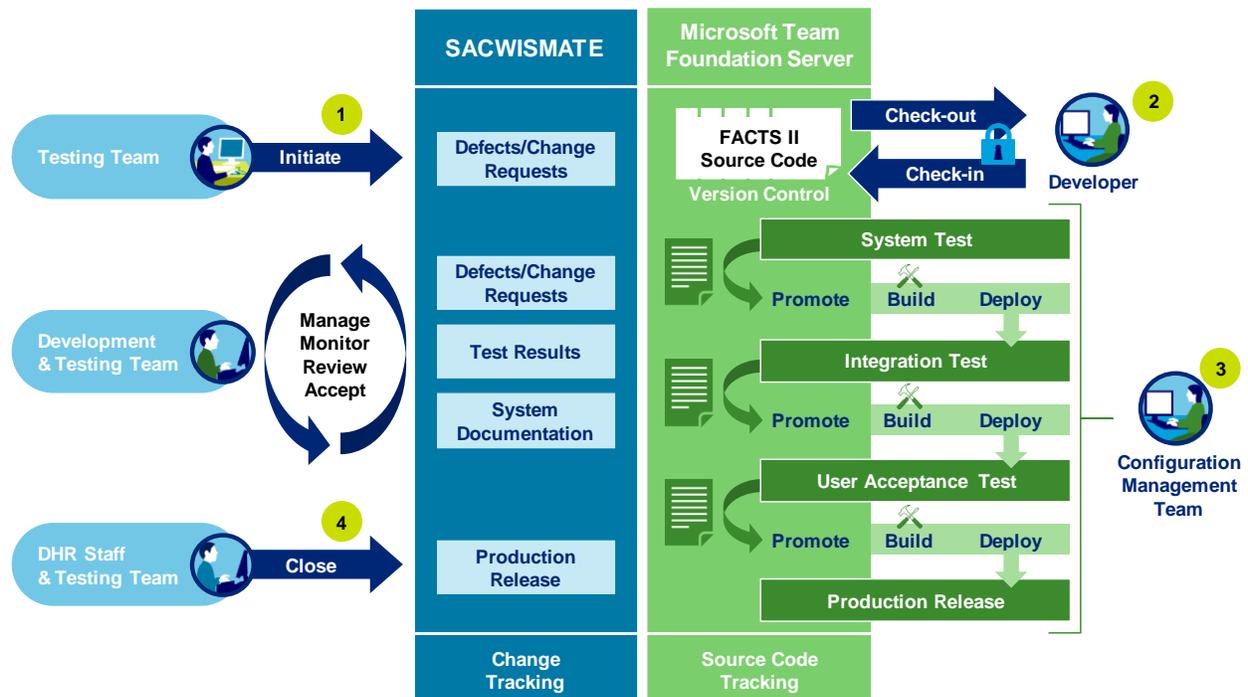
Deloitte’s unit testing approach promotes coding standards and uses peer reviews to identify, track, and resolve coding standards non-compliance improving application code maintainability.

We produce a set of well-structured system development deliverables that detail the construction of Delaware FACTS II as part of the software development phase. The objective of system development deliverables is to guide technical staff responsible for understanding and maintaining the development life cycle of Delaware FACTS II.

The Deloitte team’s approach for building Delaware FACTS II is to construct a custom solution that meets the DSCYF’s business needs and provides features that are compliant with federally compliant SACWIS requirements. We work with DSCYF to customize our development deliverables in accordance to the requirements outlined in the RFP. We acknowledge all DSCYF and Contractor roles and understand that the DSCYF gives the final sign-off of all deliverables and dictate the completion of the acceptance criteria.

Code Deployment and Migration

We use automated tools integrated with SACWISmate and Visual Source Safe to deploy and migrate fixes for defects identified during the testing phase. The following figure shows the integration and role of these automated tools.



DE_SACWIS-074_4

Figure 4.12-8. Role of Automated Tools.

Will benefit from our selected tools audit capability and traceability through improved capture of change requirements, improved tracking, and reduced development time.

A number of steps are highlighted in the illustration above that is standard and critical to the control of source code components. These steps are described below.

Step 1 – Initiate Defect/Enhancement/Change Request

Defects/Enhancement/Change Requests aka Incidents found during the testing phase is entered into SACWISmate and include description of the issue or change, affected subsystems, priority, originator information, and any other critical information required for prioritizing the change.

Step 2 – FACES.net Code Modification

All incidents to the system are grouped into releases by our development leads, based on the release plan agreed on by the Deloitte team and DSCYF. The release plan is updated on an ongoing basis based on DSCYF and project priorities and demands. The Development Team constructs and unit tests functionality according to the release schedule.

Based on their assigned incidents, developers in consultation with their leads perform impact analysis and identify source code components that require modification. Developers only modify code or update system documentation based on an incident. The developer performs “check-out” of the required code components from VSS, noting the incident that it correlates to in SACWISmate. VSS provides one secure repository for all code components. After completing modifications to the component and completing unit testing, the code is “checked-in” to VSS. Those code components checked into VSS are linked to a SACWISmate incident which, in turn, is tracked to a business requirement. VSS automatically stores the code as a new version and defaults its promotion level to the lowest level.

In consultation with their development lead, a request is made to promote the code to the next environment for further testing.

Step 3 – Promote, Build, and Deploy

Once approved and following appropriate testing, code components are promoted to the respective environment and go into the application build for that environment. Our Configuration Management Team is responsible for following detailed procedures to receive promotion requests, check approvals, check status of system documentation updates, communicate deployment start and completion times, and confirming the correct versions are pulled from VSS for the build.

Build schedules for the system and all environments is laid out in the Configuration, Build, and Release Management Plan and adhered to by the Configuration Management Team. The Configuration Management Team uses automated build scripts to expedite the build of the Delaware FACTS II application. Once a successful application is built, it is deployed to the application server and communication to appropriate stakeholders commences.

The Configuration Management Team is responsible for branching code for each new major release of the application, and maintaining the overall source code “project tree” in VSS. These are critical ongoing tasks that allow development to continue on new changes while current changes are moving their way through testing to production release.

Step 4 – Manage, Monitor, Review, Accept, Approve

From the initial incident to production release, both DSCYF and Deloitte staff is able to use SACWISmate to manage and monitor the progress of the incident. SACWISmate provides the ability for ongoing notes, file attachments, assignment and re-assignment of change requests, estimated and actual completion/resolution dates, detailed control reports, and other features. These features help the Deloitte team leads to manage changes to the system, and provide a clear view to DSCYF staff of how many incidents are outstanding, what is their status, and when they are resolved.

Development Tracking Reports

The Deloitte team brings an entire suite of development tracking reports that track the progress of the development of code components across screens, batches, reports and interfaces. Sample tracking reports are shown in the figure that follows.

Delaware Department of Services for Children, Youth and Their Families
FACTS II, RFP #07

All Use Cases - Effort Data								
	Track - 1	Track - 2	Track - 3	Track - 4	Track - 5	Track - 6	Track - 7	Total
Total Estimate	14125.64	9675	10098.12	7813	2720.7	7230.04	3317.26	54979.76
Total Projected Actual	9554.25479	4696.515	8669.917125	6782.17849	2849.637	5400.987	1824.07	39777.56
Total Variance Hours	-4571.3852	-4978.485	-1428.202875	-1030.82151	128.93695	-1829.053	-1493.19	-15202.20
Average Variance %	-32.36%	-51.46%	-14.14%	-13.19%	4.74%	-25.30%	-45.01%	-27.65%
Total UCs	37	20	39	27	17	36	12	188
Total New Screens	57	45	87	31	13	42	5	280
Total Modified Screens	58	27	42	44	39	47	13	270
Avg. Actual hrs/screen	83	65	67	90	55	61	0	72

Schedule Data								
	Track - 1	Track - 2	Track - 3	Track - 4	Track - 5	Track - 6	Track - 7	Total
Planned Dev Start	8/1/2008	8/1/2008	8/1/2008	8/1/2008	8/1/2008	8/1/2008	8/1/2008	
Actual Dev Start	9/8/2008	11/3/2008	9/8/2008	9/8/2008	9/8/2008	9/8/2008	9/8/2008	
Planned total Days for construction	238.00	238.00	238.00	238.00	238.00	238.00	238.00	
Actual construction days	190.00	150.00	190.00	190.00	190.00	190.00	190.00	
Total UCs complete	32	20	34	23	17	35	9	170
Total UCs in Track	40	22	52	28	17	45	21	225
Total time elapsed	79.83%	63.03%	79.83%	79.83%	79.83%	79.83%	79.83%	
Current Completion %	80.00%	90.91%	65.38%	82.14%	100.00%	77.78%	42.86%	75.56%

All In Progress Use Cases								
	Track - 1	Track - 2	Track - 3	Track - 4	Track - 5	Track - 6	Track - 7	Total
Total Estimate	4311	0	1608.1	1993.62	0	665	805	9382.72
Total Projected Actual	2887.9194	0	1354.7909	1681.41924	0	395.615	669.904	6989.59
Average Variance	-33.01%	0.00%	-15.76%	-15.66%	0.00%	-40.51%	0.00%	-25.51%
Total UCs in progress	5	0	5	4	0	1	3	18
Total New Screens	14	0	12	15	0	5	2	48
Total Modified Screens	20	0	0	4	0	0	6	30
Avg. Actual hrs/screen	85	0	113	88	0	79	0	90

All Completed Use Cases										
	Sep-08	Oct-08	Nov-08	Dec-08	Jan-09	Feb-09	Mar-09	Apr-09	May-09	Total
Total Estimate	0	1565.39	2506.19	3609.07	3510.14	8029.63	5431.77	11777.77	9167.08	44031.65
Total Projected Actual	0	1615.35	2617.596	3827.85	3520.9476	5830.24315	3240.3428	6330.70309	5804.9382	31172.62
Average Variance	0.00%	3.19%	4.45%	6.06%	0.31%	-27.39%	-40.34%	-46.25%	-36.68%	-29.20%
Total UCs Completed	0	13	24	19	17	20	23	27	27	157
Total New Screens	0	8	0	16	7	35	31	70	65	224
Total Modified Screens	0	24	41	36	44	26	26	28	15	216
Avg. Actual hrs/screen	0	50	64	74	69	96	57	65	73	71

All Projected Use Case Completion									
	9-Feb	23-Feb	9-Mar	23-Mar	6-Apr	20-Apr	4-May	18-May	1-Jun
Total Estimate	2727.06	4881.33	2639.36	3521.31	5648.2	2164.77	4414.1	4627.48	2983.8
Total Projected Actual	1495.8	4081.03785	1895.2138	1849.788675	1974.1366	1613.63369	3120.2437	2631.9929	2079.8387
Average Variance	-45.15%	-16.39%	-28.19%	-47.47%	-65.05%	-25.46%	-29.31%	-43.12%	-30.30%
Projected Completed UCs	6	12	15	11	10	10	10	13	10
Total New Screens	8	22	18	17	28	29	21	27	23
Total Modified Screens	18	16	16	8	7	4	21	13	1
Total Screens being delivered	26	38	34	25	35	33	42	40	24
Total screens per plan	18	29	30	17	18	32	44	35	24
Build Variance	8	9	4	8	17	1	-2	5	0
Aggregate screens delivered	26	64	98	123	158	191	233	273	297
Aggregate screen per plan	18	47	77	94	112	144	188	223	247
Aggregate Variance	8	17	21	29	46	47	45	50	50
Avg. Actual hrs/screen	58	107	56	74	56	49	74	66	87

Include	Iteration	Track	Use Case	UC ID	Status	Total						
						Est.	Sec.	New Est.	Max %	Actual Hrs	% Comp.	Est. to compl.
Y	Iteration 13	Track - 1	IV06 - Record Safety Assessment (UC123)	137	In Progress	712.20	169.36	542.84	73.00	359.50	47.00	516.90
Y	Iteration 8	Track - 1	IV06a - SIU Safety Assessment	513	In Progress	827.10	110.48	716.62	86.00	318.20	49.00	563.02
Y	Iteration 8	Track - 1	IV13 - Record Classification	512	In Progress	832.50	87.12	745.38	90.00	67.50	15.00	579.00
Y	Iteration 8	Track - 1	IV20 - Maintain Investigation Overview	445	In Progress	1399.00	114.84	1284.16	92.00	625.00	62.00	960.76
Y	Iteration 8	Track - 1	IV23 - Record IPA	524	In Progress	520.20	182.20	338.00	64.00	56.00	16.00	268.24
Y	Iteration 5	Track - 3	CM01d-Record Adjudication/Disposition Information (UC394)	181	In Progress	145.70	0.00	145.70	100.00	73.00	52.00	135.94
Y	Iteration 18	Track - 3	JJ13a - Link Unlink CFDM	522	In Progress	286.00	3.00	283.00	99.00	94.00	40.00	228.99
Y	Iteration 14	Track - 3	JJ30 - Manage Special Caution Alert (UC379)	202	In Progress	204.40	9.00	195.40	96.00	65.50	31.00	171.79
Y	Iteration 14	Track - 3	JJ33 - Manage Youth Disciplinary Reports (UC378)	203	In Progress	566.10	19.20	546.90	97.00	121.00	28.00	433.49
Y	Iteration 13	Track - 3	JJ38- Manage Release (UC385)	201	In Progress	405.90	6.00	399.90	98.00	300.50	75.00	384.52
Y	Iteration 10	Track - 4	FM04a-Maintain Subsidy Appeal Information (UC435)	231	In Progress	25.42	0.00	25.42	100.00	19.00	82.00	23.12
Y	Iteration 3	Track - 4	FM08- Maintain Service Ceilings (UC480)	217	In Progress	1470.00	70.08	1399.92	95.00	735.00	58.00	1224.51
Y	Iteration 9	Track - 4	FM15- Determine Broker Day Care Case Services (UC457)	230	In Progress	341.30	28.99	312.32	100.00	180.00	64.00	290.58
Y	Iteration 1	Track - 4	FM20a Maintain Commissary Requests (UC462)	424	In Progress	156.90	12.80	144.10	91.00	2.00	1.00	143.21
Y	Iteration 1	Track - 6	CF68 - Maintain Transitional Surveys	441	In Progress	665.00	44.00	621.00	93.00	242.00	60.00	395.62
Y	Iteration 2	Track - 7	CF41 - Log-In and Password Change (UC600)	348	In Progress	156.00	9.60	146.40	100.00	60.00	44.00	138.62
Y	Iteration 13	Track - 7	TF01-Address Validation (UC701)	364	In Progress	162.00	0.00	162.00	100.00	42.00	38.00	132.40
Y	Iteration 6	Track - 7	TF05-GIS (UC705)	357	In Progress	487.00	4.00	483.00	99.00	188.50	51.00	398.88

DE_SACWIS-337

Figure 4.12-9. Sample Development Tracking Reports.

Incident Tracking in SACWISmate

The Incident Tracking module in SACWISmate supports the change control process and captures the reporting of suspected coding problems, hardware problems and batch discrepancies (defects) or functionality change requests (enhancements) that result from testing the system and resolving issues.

Incidents in this context relate to any imposed change to the SACWIS, either a defect or enhancement. These may originate from project staff during development and testing, DSCYF staff and users, interface testers, and through the monitoring of batch reports.

Information captured electronically in the Incident Tracking module includes a one-line name of the incident, who discovered the incident, when the incident was discovered, the severity of the incident, the environment, and the affected programs.

The tool also tracks whether or not the requested change should be escalated to a project review board. The figure below shows details on the information captured on Incidents.

The screenshot shows the Microsoft Access interface for the Incident Tracking module. The title bar reads "Microsoft Access - [Incident Tracking]". The menu bar includes File, Edit, View, Insert, Format, Records, Tools, Window, and Help. A search bar on the right says "Type a question for help". The main window has a header "Incident Tracking" with buttons for "New Incident" and "Go To Incident...". The form displays the following fields:

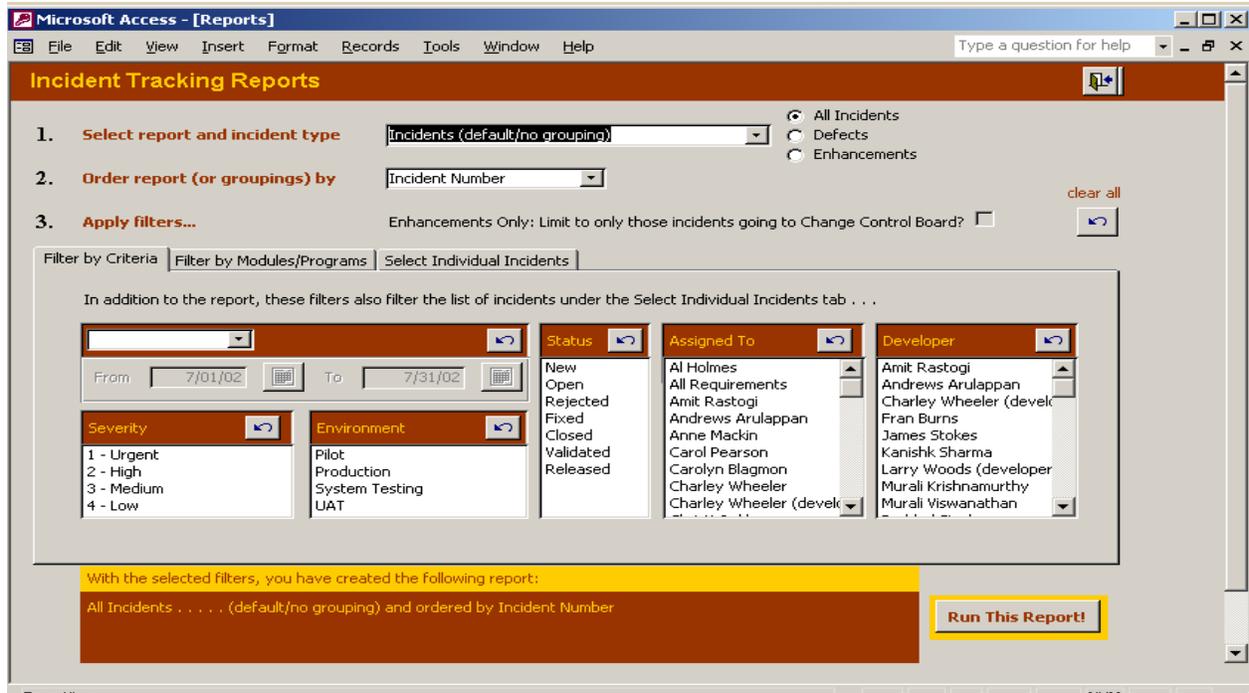
- Incident Number: 99
- * Incident Summary: Incident Name here
- Status: New
- Incident Details: Screenshot, Development, Retesting, Status History
- * Detected By: Tim Dussinger
- * Detection Date: 1/29/2002
- * Incident Type: Enhancement (selected), Defect
- * Severity: 1 - Urgent
- * Need for review by Change Control Board: No
- * Environment: Pilot
- Test Script: (empty)
- * Program/Screen ID: AP0051C
- Program Name: Placement Payment
- Functional Module: Financial Management
- SubModule: Accounts Receivable
- * Description/Comments Log: >>>> Tim Dussinger 1/29/2002 8:07:52 PM: testing
- Assigned To: testing Coordinator
- * asterisk indicates mandatory field
- Delete Incident button

DE_SACWIS-066

Figure 4.12-10. Incident Tracking.

The Incident Tracking Reports module of SACWISmate also allows reporting of incidents in multiple variations. As shown below, reports of incidents by type (defect versus enhancement), functional module, status, priority, and more are available. The reports give project management a quick snapshot of the state of the application. In addition to these summary reports, functional team leads can use the

robust reporting functionality to count incidents and to print details by functional areas, specific screen or report and current assignment. All of these report options are part of the SACWISmate tool that we bring to DSCYF, and if necessary, we develop custom queries and reports for specific management and reporting.



DE_SACWIS-067

Figure 4.12-11. Incident Tracking Reports.

Deloitte's Information Technology Service Experience

Human Services agencies face challenges today that lie not only in program delivery, but also in the ability to manage services, to deliver service to constituents, and to demonstrate that they are accountable to the changing marketplace. We have helped HHS agencies, like DSCYF, develop and maintain innovative solutions to meet these challenges head-on. We have provided IT consulting services in support of the systems integration, development and implementation of the **highest number of federally assessed child welfare systems to date**. The size and scale of the work we do for our national HHS clients provides DSCYF with the confidence that you are working with a firm that has the breadth of corporate experience and the deep bench of resources to address the most complex business and technology challenges that you may encounter.

Our knowledge and experience of implementing our core methodologies and tools, PMM4, Playbook and SACWISmate across all our integrated children's service projects nationwide has not only been successful but also helps focus on continuing to enhance the methodology with experiences and lessons learned from our clients and project base to improve communication, deliberate management controls, transparency and accountability. Our development component of our Playbook systems development

methodology has been used on every child welfare project and SACWIS solution since the beginning, 1994. While the methodology has been refined over the years and adapted to meet State specific standards, the core foundation is the back bone of our delivery track record and quality solution.

Anticipated Project Team Members' Experience with Proposed Methodology

The following matrix summarizes our proposed key staff's experience in the proposed methodologies, and services relevant to the Delaware FACTS II solution. More detailed profiles on each key staff member can be found in the "Key Staff Profiles" subsection of *Section 10, Resumes of the Bidders Proposed Staff*, along with the data points requested to be included with each individual's resume in the RFP. All sub contractor staff on the Deloitte team receive Deloitte tools and methodology training before the project begins in preparation to "hit the ground" running.

Proposed Methodology and Services	Project Team Members' Level of Expertise in Meeting RFP Requirements					
	Nicole Fuller	Ajit Kulkarni	Prabhakar Arulraj	Kim Buchheit	Harvey White	Michael Nazareth
FACTS II Playbook	✓	✓	✓	✓		✓
SACWISmate	✓	✓	✓	✓		✓
Project Management	✓	✓	✓	✓	✓	✓
User Adoption and Training	✓	✓	✓	✓	✓	✓
Application Modifications	✓	✓	✓	✓	✓	✓
System Implementation	✓	✓	✓	✓	✓	✓
IT Strategy and Process Improvement	✓	✓	✓	✓	✓	✓
Enterprise Service Oriented Architecture	✓	✓	✓	✓	✓	✓
System Integration Services	✓	✓	✓	✓	✓	✓

Table 4.12-3. Deloitte's Project Team Members' Experience in Meeting RFP Requirements.

Approach to Risk Assessment, Mitigation and Management

Deloitte's Risk Management approach prescribed by the Delaware FACTS II Playbook will help to quickly identify potential issues, and initiate appropriate corrective actions before they become critical. We recognize that project problems and issues inevitably arise during the project. Good management involves accurate and timely problem identification, escalation and resolution. This approach is described in *Section 4.5*. The implementation of key processes based on our proposed tools above help create uniformity of approach

and reduce human errors in code management. DSCYF benefits from use of these tools by being able to see the status of changes, approving test results and production releases, and overall traceability of project change requests. In closing, our Delaware FACTS II Playbook methodology was the foundation of our successful deliveries on our large-scale SACWIS projects. Specifically, our project management methodology supports a structured, yet flexible approach to monitoring work tasks, resource allocation, issue resolution, risk mitigation, change control, and the configuration management controls.

4.12.1 Associated Deliverables

RFP reference: 6.12.1 Associated Deliverables, Page 51

The following deliverables are required during the System Development Phase:

- Application Standards;
- Code Releases;
- Migration Plan;
- System Backup and Recovery Plan;
- Configuration, Build, and Release Management Plan;
- Integrated Application Development Environment (IADE) Design Document; and
- The application system, the development environment, all programs, and the documentation required to support and operate the system.
- For each build the State will be provided with copies of the current system builds. They should include source code, scripts, and database schemas.
- The State should be involved in the vendors change control process

Throughout the Systems development phase the vendor must provide the State with visibility into their development processes

The outcome of the System Development phase is the creation and submission for DSCYF approval the following deliverables:

- Application Standards
- Code Releases
- Migration Plan
- System Backup and Recovery Plan
- Configuration, Build, and Release Management Plan
- Integrated Application Development Environment (IADE) Design Document
- The application system, the development environment, all programs, and the documentation required to support and operate the system
- For each build DSCYF is provided with copies of the current system builds. They should include source code, scripts, and database schemas.

Throughout the Systems development phase the Deloitte team will provide DSCYF with visibility into our development processes.

