Prepared for:

Centers for Medicare & Medicaid Services

CMS Alliance to Modernize Healthcare
Federally Funded Research and Development Center

Consumer Information and Insurance Systems Group

Task Order No. HHSM500T0015

Supplement to FFRDC Technical Assistance Report – Commonwealth of Massachusetts Health Connector

Not for Public Release

Version 1.0

February 25, 2014

Record of Changes

Version	Date	Author / Owner	Description of Change
1.0	February 25, 2014	The MITRE Corporation	Final for delivery

Table of Contents

List of Tables

1. Introduction

1.1 Background[1]

In order to meet the Affordable Care Act's (ACA) timelines and vision for a "single, streamlined eligibility and enrollment process", the Commonwealth of Massachusetts developed a "single project approach" to coordinate development, procurement, and implementation of Health Insurance Exchange (HIX) and Integrated Eligibility System (IES) technology design and implementation. The overall project is now referred to as the "Massachusetts HIX/IES Project".

To implement the information technology (IT) system that supports ACA implementation, the Commonwealth was awarded an Early Innovator Cooperative Agreement from the Centers for Medicare & Medicaid Services (CMS) beginning in February 2011. Massachusetts has been subsequently awarded Center for Consumer Information and Insurance Oversight (CCIIO) Establishment funds, as well as funding from Center for Medicaid & CHIP Services (CMCS) in December 2011 in the form of an Implementation Advance Planning Document (IAPD).

The Health Connector is independently operated by the Commonwealth Connector Authority (CCA) and is intended to serve as the common front door that citizens use to gain access to affordable health insurance and Medicaid. The Massachusetts Executive Office of Health and Human Services (EOHHS) administers, under the MassHealth Program, the state's Medicaid Program.

The University of Massachusetts Medical School (UMMS) provides contract acquisition and contract management services for the systems integrator (SI) and independent verification and validation (IV&V) vendor contracts. The MA HIX/IES tri-party agreement defines the roles, responsibilities, obligations, and governance structure by and between the MA HIX/IES entities—CCA, MassHealth, and UMMS.

In July 2012, the Commonwealth awarded the SI vendor contract to CGI (as the prime contractor).

1.2 ACA Implementation in Massachusetts

The ACA requires that states implementing a Marketplace first determine eligibility for Medicaid based on current Modified Adjusted Gross Income (MAGI). If the applicant is ineligible for Medicaid, the state must then determine eligibility for subsidized coverage, also known as an Advance Premium Tax Credit (APTC), before offering unsubsidized coverage through a process that allows applicants to compare plans and enroll in the ones they prefer.

The Commonwealth has chosen to build a single web-based portal, the MA HIX/IES, which will make eligibility determinations and allow residents to shop for and enroll in plans offered by the Massachusetts health insurance marketplace, the Massachusetts Health Connector, and MassHealth.

Applicants considered eligible for Medicaid are enrolled in the MassHealth programs through a Medicaid Management Information System (MMIS). Applicants eligible for subsidized coverage

[1] Information in the *Background* is based on input from Commonwealth staff.

and those who want to purchase unsubsidized plans may shop for and enroll in plans through the Massachusetts Health Connector.

In addition to determining program eligibility and supporting enrollment, the MA HIX/IES was to offer such key marketplace functions as plan management, financial management, Small Business Health Options Program (SHOP), customer service, outreach, and reporting.

1.3 MA HIX / IES Implementation Issues

The Commonwealth has faced and overcome significant challenges over the development life span to build the new HIX/IES. Unfortunately, the HIX/IES website was not fully operational on October 1, 2013, the federally mandated start of the initial health insurance marketplace open enrollment period in accordance with the ACA. The current MA HIX/IES system suffers from technology infrastructure and data stability problems. The limited functionality implemented to date demonstrates significant usability and performance issues. To ensure that Massachusetts residents who wish to obtain health insurance coverage can do so, the Commonwealth has implemented manual workarounds for many processes. Taken together, these workarounds have caused cascading effects across the Health Connector and MassHealth back office functions, including surges in Call Center activity and overloading of Health Connector and MassHealth staff. As of the date of this report, the Commonwealth has significant concerns with:

1. The quality of the HIX/IES system that has been deployed to date
2. The integrity of the underlying data model
3. The SI vendor's ability to deliver much needed core functionality in the near term

1.4 Scope of Work

The Commonwealth requested technical assistance from CMS, its federal partner, in evaluating the construction of the HIX/IES. CMS requested IT and systems engineering assistance from The MITRE Corporation (MITRE), operator of the CMS Alliance to Modernize Healthcare (CAMH) Federally Funded Research and Development Center (FFRDC). MITRE has developed substantive knowledge and understanding of the Commonwealth's systems development life cycle. MITRE has also provided technical guidance and assistance to the Commonwealth through its engagement with CMS in supporting and assessing various stages of development through gate reviews.

The scope of this engagement was to conduct a systematic review of the IT functions of the Commonwealth's HIX/IES. Specifically, the Commonwealth requested that this review include:

- A technical review that assessed the HIX/IES IT solution. This technical review included, but was not limited to, elements of:
  - Project structure, roles, and responsibilities
  - Project planning and execution
  - IT governance/software life cycle processes
  - Technical elements

- An approach for recovery that included:
  - Recommendations on near-term (30–45 days) and long-term fixes required to help restructure the HIX/IES IT organizational and governance processes

This engagement was not intended to constitute a comprehensive programmatic or technical review, but rather provide objective, independent insights into those areas where CMS and the Commonwealth noted their most critical concerns.

1.5 Purpose

The purpose of this *Supplement to FFRDC Technical Assistance Report* is to present supplemental information regarding MITRE's independent review and assessment of the Massachusetts HIX/IES project. The FFRDC Technical Assistance Report referred to in this document (the "Technical Assistance Report" or the "main report") addresses the Commonwealth of Massachusetts Health Connector. This supplement to the main report provides additional information about the technical assessment, findings, and recommendations concerning the Commonwealth's implementation of their healthcare marketplace website.

As noted in the Technical Assistance Report, the Commonwealth remains responsible for the disposition of recommendations presented here and for decisions related to program direction, contracts, and funding in working collaboration with CMS.

1.6 Approach

This supplement provides additional information to support the results of MITRE's assessment of the project to implement the Commonwealth of Massachusetts's HIX/IES, the website that Massachusetts residents will use to enroll in a qualified health plan or Medicaid. CMS and the Commonwealth of Massachusetts asked MITRE to perform this assessment and to provide the additional detail documented here. Summary results of the assessment are contained in the Technical Assistance Report. This supplement is not intended as standalone document and should be read in conjunction with the main report.

1.7 Document Organization

In each of the following sections of this supplement, MITRE provides additional background, observations, or other details to support the findings and recommendations noted in the Technical Assistance Report. Section 2 addresses the assessment approach, and Sections 3 and 4 address the findings and recommendations, respectively. MITRE notes wherever there is no additional information available.

2. Detailed Assessment Approach

This section provides additional detail about the approach MITRE used to conduct the assessment. As described in Section 1.6 of the Technical Assistance Report, the assessment focused on four major areas:

- Project structure, roles, and responsibilities
- Project planning and execution
- IT governance/software life-cycle process
- Technical elements of the project

The following subsections address each of these topics in detail.

Note: Section 2 of the Technical Assistance Report documented a number of concerns in each of these major areas that were identified prior to the assessment. This section presents no additional information regarding the listed concerns.

2.1 Project Structure, Roles, and Responsibilities

The purpose of this focus area is to examine the project's organizational structure. The assessment sought to determine if the minimally required technical and management roles were identified, if the responsibilities for those roles were clearly defined and communicated, and if the team well understood the accountability for those roles. The independent assessment also determined whether there were role and skill gaps.

2.2 Project Planning and Execution

The assessment of this area focused on determining if the project's needs and expectations, as well as the IT management processes for HIX/IES development, were identified and well understood. The assessment looked at whether a well thought-out management plan and well-developed schedule were laid out so that the team could establish common goals, communicate those goals with each other, measure progress against those goals, and manage changes. The assessment also evaluated how the execution team utilized effective controls to ensure that all project work was performed effectively, including planning for:

- **Scope –** Business needs, requirements, deliverables, constraints, and work breakdown structure
- **Schedule –** Activities schedule and project milestones
- **Quality –** Quality measurement and control
- **Communication –** Communication types, channels, and reporting
- **Risks –** Methods to identify, evaluate, monitor and control risks, risk mitigation, and contingency planning

2.3 IT Governance / Software Life-Cycle Process

This area's assessment focused on the use of governance and life cycle processes. Topics covered in this assessment include:

- **Requirements Management** – To verify whether the life cycle of a requirement, including traceability and sign-off processes, is clearly understood and implemented
- **Change Management** – To understand the processes and best practices used to manage impacts of change during development. This includes an analysis of the defect management techniques and methods as well as communications and impact assessments methodology.
- **Build/Release Management** – To verify whether existing configuration management tools and processes were appropriately implemented, fully communicated across life cycle boundaries, and effective
- **Design/Implementation** – To determine if the design were complete and appropriate for implementation
- **Testing** – To verify that there were sufficient tools, procedures, processes, schedules, data, and handoffs between stages of testing for packaging and release

2.4 Technical Elements

The assessment of technical elements included sessions on the following areas, for the reasons described:

- **Architecture –** To review the systems architecture to discover if it were built on sound principles and decisions, including:
  - Assess architectural decisions to determine how they enable or restrict a system's ability to meet its architecturally significant requirements
  - Assess if the architectures were defined early in the project life cycle and conformed to defined standards and best practices

  The most significant benefit of architectural evaluation is to assure stakeholders that the candidate architecture is capable of supporting the current and future business objectives—specifically, that it can meet its functional and nonfunctional requirements. A system's quality attributes, such as performance, availability, extensibility, and security, are a direct result of its architecture. An evaluation of the architecture, while it is still a candidate specification, can greatly reduce project risk and minimize the need for introducing quality into the system late in the game.
- **Software Design –** To review the software design process to understand:
  - Whether the software components and their expressions rigorously capture all facets of the life cycle, from business requirements to functional decompositions and interdependencies
  - Whether the requirements were properly represented in the architecture, service offerings, data representation, and infrastructure

- **Data –** To review the data structures and content to:
  - Determine if there were traceability from a business data model to the logical and physical data models
  - Determine whether sufficient rigor were imposed to understand the data required to represent the business needs
  - Ensure that the system has data integrity during the stages of intake, transmission, persistence, and transformation
  - Ascertain whether the development and testing processes created appropriate complexities in data to support verification during test scenarios
- **Integration –** To gain insight into the techniques used to integrate and verify that a broad range of technical components work properly together.
- **Infrastructure –** To understand whether the hardware, software, network, and platform choices were:
  - Appropriate for the business needs
  - Properly sized to support the expected workload
  - Sufficiently redundant to minimize failures, manage loads, and scale for growth

2.5 Additional Review Topics

The technical review also included sessions on various topics. There were unique reasons for technical review in each of the following areas:

- **System Architecture –** To understand whether architectural processes were appropriate or were impeding the development of system functionality.
- **Account Creation –** To understand the difficulties that had been occurring with the account creation process.
- **Data Access / Integrity –** To understand the data architecture employed and the issues that the system had been experiencing.
- **Testing Procedures / Protocols / Tactics / Strategies –** To understand the testing process and its implementation.
- **Requirements –** To understand how the requirements are being managed and traced through the design to use cases, technical implementations, code, and testing.
- **Portal Layer** (application data entry) **–** To understand how the portal was structured for application intake and how well it is functioning.
- **Eligibility Determination –** To understand how the system has implemented eligibility determination and the areas that need further work.
- **Shopping –** To understand how the system has implemented shopping and the areas that need further work.
- **Build Management** – To understand the process for creating and applying a new release.

- **Environments** – To understand how the environments are set up and whether they were impacting performance.
- **Project Management** – To understand how the project is being managed and the processes followed.

3. Detailed Findings

This section presents MITRE's findings based on the assessment. Findings address the four major areas of focus:

- Project structure, roles, and responsibilities
- Project planning and execution
- IT governance/software life-cycle processes
- Technical elements

Each focus area consists of a number of topic areas.

3.1 Project Structure, Roles, and Responsibilities

This subsection contains additional findings and or observations concerning issues with the structure of the project, including roles and responsibilities of all organizations involved.

3.1.1 Commonwealth Team

There is no additional information in this area.

3.1.2 SI Vendor Delivery Team

There is no additional information in this area.

The following subsections (3.1.2.1–3.1.2.7) contain more detailed findings on the SI vendor team.

3.1.2.1 Project Management

There is no additional information in this area.

3.1.2.2 Business Function Integration Leadership

There is no additional information in this area.

3.1.2.3 Technical System Integration Leadership

The following additional observations/findings support the summary findings found in subsection 3.1.2.3 of the Technical Assistance Report:

- The project has an application Delivery Manager and a Solution Architect. It appears that the Delivery Manager plays the role of an Integration Manager, while the Solution Architect appears to play multiple roles.
- There is no Technical Lead Architect (Chief Architect) acting as a peer to the Lead Business Architect to ensure the successful implementation of the technical requirements. This is a critical gap. In addition, there is no dedicated portal architect who can enforce appropriate configurations, coding standards, and design elaborations to prevent a "design-on-the-go" approach. This role would be expected if the architectural construct

relies largely on a portal-heavy implementation—that is, an implementation where the portal layer performs considerable processing.

3.1.2.4 Data Architecture Management

There was no apparent logical data model developed for this effort. While it would be good to have such a model, it is not necessarily critical if a physical model can be developed from a previous version.

3.1.2.5 Quality Assurance

There is no additional information in this area.

3.1.2.6 IT Process Leadership

The following additional observations/findings support the summary findings found in subsection 3.1.2.6 of the Technical Assistance Report:

- The SI vendor has defined coding standards for handling errors, but it appears that these standards are neither used nor enforced.

3.1.2.7 User Interface Development Resources

There is no additional information in this area.

3.2 Project Planning and Execution

3.2.1 Project Planning and Execution

The following additional observations/findings support the summary findings found in subsection 3.2.1 of the Technical Assistance Report:

- The project baselines were never clearly understood, and never clearly enforced and managed. Project rebaselining took place on schedules without sufficient understanding of risks, resource availability, and necessary adjustments.
- There were multiple, unreconciled schedules. These schedules were unrealistic given the lack of roles and skill depth in the project teams. Despite repeated rescoping efforts, the inability to manage scope creep and a lack of understanding of the impact of that creep (which would have been available through a strong change and impact management function) has resulted in late delivery.

3.2.2 Scope Management

The following additional observations/findings support the summary findings found in subsection 3.2.2 of the Technical Assistance Report:

- Added scope and the inability to manage the expanded scope have compromised the project's credibility and quality. The worker portal is one example of an area where scope creep has impacted performance. This additional work (approximately $20–25M) was

added to the SI vendor's workload in early spring 2013 and may have impacted resources that would have been available to complete the rest of the website development.

- The program continues to experience scope challenges with over 400 changes currently identified. While most of these are still on hold, the magnitude demonstrates the complex challenges as well as expectations that additional work will be needed.

3.2.3 Work Breakdown Structure

There is no additional information in this area.

3.2.4 Work Alignment

There is no additional information in this area.

3.2.5 Schedule

There is no additional information in this area.

3.2.6 Quality Assurance

There is no additional information in this area.

3.2.7 Communications

There is no additional information in this area.

3.2.8 Risk Management

The following additional observations/findings support the summary findings found in subsection 3.2.8 of the Technical Assistance Report:

- The Project Management Office Risk Manager typically sends out periodic queries to risk owners requesting updates. It does not appear the risk register is being used because few risks have been updated.
- It is not clear that the SI vendor has a true risk management process because there does not appear to be sufficient monitoring or follow up.

3.3 IT Governance / Software Life-cycle Process

3.3.1 Requirements Management

The following additional observations/findings support the summary findings found in subsection 3.3.1 of the Technical Assistance Report:

- The project team is using Rational Composer to trace customer requirements to business requirements and high-level business design, and then to trace them to downstream test cases. The requirements, however, are not fully traced to the technical life-cycle elements implementing the business requirement. The technical component designs have been traced to the wireframes found in the use cases, but the implementing elements are not apparent when looking at the use cases/wireframes. There is no traceability from the

requirements to the code. As the requirements evolved, the wireframes were not kept current.

- While there is evidence of traceability to test cases, it is not clear that the test cases were written to test all the permutations and combinations of data and exception conditions. Currently, it is very difficult to understand which requirements have been fully implemented.

3.3.2 Change Management

There is no additional information in this area.

3.3.3 Build / Release Management

The following additional observations/findings support the summary findings found in subsection 3.3.3 of the Technical Assistance Report:

- As new work scope is identified, the SI vendor assigns the work to a worker. The worker will then either modify existing code or develop new code. When the SI vendor believes the code is ready, the SI vendor indicates this in the subversion tool. Once checked in, the code is then incorporated in the next build. The tool has processes to help identify and de-conflict code when two developers are simultaneously working on the same component.

3.3.4 Design / Implementation

The following additional observations/findings support the summary findings found in subsection 3.3.4 of the Technical Assistance Report:

- The SI vendor held separate war room meetings between the business architects and the technical architects to attempt to flesh out more of the details. The war room meetings did not include Commonwealth subject matter experts (SME), and the meetings were not fully documented. Since many of the implementation issues appear to be how the requirements are interpreted or implemented, it is clear that the process for translating and implementing the requirements needs to be improved.
- The SI vendor currently has a number of processes documented, but these processes need to be more rigorously enforced. To improve the creation, maintenance, and enforcement of project and system engineering processes, it would be useful to have process experts as part of the project team. Process experts would ensure conformance to process standards, for example CMMI®, through verification procedures or independent process audits. Ultimately, process conformance helps the Project Manager manage and mitigate risks as the project executes.
- The design is still under discussion for which data to retain and which to purge when an applicant shifts from requesting a subsidy to not requesting a subsidy.
- The SI vendor provided an overview of its coding standards, but the project does not appear to be following those standards. For example, the standards call for error handling, but error handling does not appear to have been fully implemented in the system. Errors

do not appear to have been fully assessed and trapped. In addition, it was not clear that sign-offs were required prior to transition to subsequent steps within the life cycle.

3.3.5 Testing

The following additional observations/findings support the summary findings found in subsection 3.3.5 of the Technical Assistance Report:

- Since there is no traceability from the requirements to the technical components implementing the requirements, it is unlikely that all the technical implementation aspects of the design have been tested. This lack of traceability compromises any assessment to determine robust testing of integrated components.
- The SI vendor identified approximately 4,300 test scripts. Only 50 percent of these test scripts were run for Release 1.0 (R1.0) functionality, and about 68 percent passed. Of the tests that were run, the SI vendor reported that approximately 75 percent were unit tests. Accordingly, less than 600 integrated system tests were run, and it was unlikely that more than 400 passed.

  The IV&V contractor was the User Acceptance Test (UAT) lead. The IV&V contractor reported, even though the system did not meet UAT entry criteria, they had executed approximately 700 test scenarios against in-work versions of release R1.0, and approximately 41 of the tests passed. It was noted that the version tested in UAT was several iterations older than the one that actually deployed as R1.0, but no UAT was completed on the deployed version.
- The rules engine does not appear to have been tested using a broad set of test cases focused on this integrated component. For a system of this magnitude, integrated component testing, where possible, promotes confidence that the system will be able to function in an end-to-end environment.
- The Commonwealth did develop a Semi-Automated Contingency Tool (SACT)[2] that may be able to perform testing on the rules themselves, but there does not appear to be a complete set of test cases, with appropriate test data, ready to run through this tool. Instead, the only testing that was developed appears to use end-to end-scenarios that are more aligned with UAT-type testing and do not allow for quick identification of problems, particularly when there have been problems with preceding steps.
- Virtually all of the test cases are documented as having been tested the normal path, although a few did have variants to evaluate alternate paths. There has been no systematic testing of negative scenarios. Although some ad hoc testing of negative cases may have occurred, it is not clear how comprehensive this was since test conditions and results were not recorded, unless defects were noted.
- There does not appear to have been any significant performance testing completed prior to the 10/1/2013 go live date. The SI vendor has recently started some performance testing, but it is still in its early stages.

[2] The SACT was developed by extracting the business rules engine from the system and allowing the user to identify the data to be submitted for a determination.

- Test scripts for assessing the success of defect fixes in a given release appears to be lacking. Regression testing of the corrected defects in subsequent releases is not apparent.
- Given the considerable amount of ad hoc testing, it appears that formal testing and the development of test scripts to support formal testing as well as regression testing has been dropped.
- No true UAT is being conducted on Release 1.1. The testing for Release 1.1 takes place in the development/System Integration testing environment using joint Commonwealth SME/SI developer testing. No UAT environment currently exists. Testing is currently being conducted in System Integration Testing (SIT) environment, which creates challenges in conducting efficient testing and a smooth transition to production.

3.4 Technical Elements

3.4.1 Architecture

The following additional observations/findings support the summary findings found in subsection 3.4.1 of the Technical Assistance Report:

- A considerable amount of processing occurs in the presentation (portal) layer. The business process orchestration engine does not appear to be managing the presentation layer.
- The presentation layer appears to be directly accessing the database (via spring / hibernate), and does not appear to employing data abstraction.
- Applicable plans and the rates for plans are obtained through calls to carriers.
- The HIX/IES's functional architecture, process decompositions and interactions, and systematic mapping of those processes to a business service component in the SOA are not fully documented. There is a lack of alignment between the business and technical services throughout the software design and development processes. Establishing appropriate service level expectations between the provider service and the consumer service, using the Service bus, has been deficient. The SI vendor was unable to show a master catalog of all the services, but instead provided one for only the external services.
- Data abstraction—commonly achieved between the data layer and the portal (presentation) layer—through service orchestration for access, security, aggregation, transformation, transaction support, and to broker business information, is insufficient and has caused portal scalability and data integrity issues. The portal layer does not include data isolation and does not take advantage of data abstraction. In addition, the transaction management functions are not accomplished through enterprise components. This has created performance issues for the system and end user. Unless these architectural issues are rectified, it will be difficult to maintain the system.
- The HIX/IES's current technical platform includes tools that are capable of supporting SOA as a design principle. SOA realization occurs through technical implementations that provide agile service composition, discovery, and virtualization by keeping the alignment between the business architecture and infrastructure flexible and continuous. This usually occurs through a process orchestration layer as well-defined service

component architecture and a set of entity services. Successful implementation of SOA requires run-time governance of service elements, but the HIX/IES's technical implementation does not always follow these principles.

- The incomplete implementation of SOA has resulted in an architecture that has not realized such quality attributes as performance, availability, extensibility, scalability, and maintainability. The original production system was not linearly scalable. Linear scalability was added on 12/24/13.
- At this time, it appears that the portal layer includes the functionality to accomplish the basic application intake functions, but the process is prone to errors that frustrate the end user. Neither the Program Determination nor the Notices portions of the system appear to be fully functioning. Testing of the system was not completed for any of these areas. The system is marginally operational. Many issues could be addressed if user-friendly error messages were provided through the portal to the applicants.
- The system uses the Oracle Policy Automation (OPA) rules engine and the Expressions notices generation system. During application intake, a session is opened and information is held within the session until a page transition occurs. At that time, the data is written into the application tables within the database. When the user certifies the application is ready for Program Determination, the system transfers the information to the member tables. The system is currently "gating" applications that request financial assistance from receiving a Program Determinations and Shopping. Only individuals who do not request financial assistance can proceed through Program Determination and Shopping.
- The Program Determination module still needs considerable work. Many areas are complex because they encompass more program determination categories and required significantly more testing. The extent of issues within this area appears to be unknown, because only about 50 percent of the multi-applicant (couple and family) scenarios passed during joint testing. These scenarios represent less than a quarter of the scenarios that need to be executed.
- Program Determination is done using the OPA rules engine, with no real data manipulation expected before data is passed to the rules engine. The rules engine is not processing any applications where subsidized health care is requested. Originally, Release 1.1 was going to expand Program Determination to cover simple single individual applications, but temporary changes to Program Determination codes are needed. The other portions of the eligibility determination rules for couples and complex family structures are still undergoing testing.
- The Notices portion also appears to need additional work. Notices are being implemented so that combined notices are generated asynchronously, but this functionality is not currently enabled. Although the 60 scenarios for simple individual applications appear to have made it through testing, significant work is necessary to verify that the notices are generated correctly for more complex family structures.
- The system uses a round robin, load-balancing server methodology affinity for each session, which results in the loss of session data during server failures.

3.4.2 Software Design

The following additional observations/findings support the summary findings found in subsection 3.4.2 of the Technical Assistance Report:

- The failure to incorporate appropriate error handling has resulted in a default screen error code that is meaningless to the user, creating an inability to proceed to the next step. In some cases, the system does not provide the user a way to gracefully exit a portion of the system. The user is therefore stuck, unable to proceed and unable to exit, without logging off.
- Design components such as dependencies, error handling, default values, and mappings between dependent and peer components, are missing. Software specifications are not complete; they will not support appropriate code.
- The business rules engine as a service appears to be computing correctly, indicative of a rigorous design. It does not appear, however, that there has been sufficient testing to justify confidence in the complete accuracy of the engine.
- The project has documented a number of non-functional technical requirements for capacity, scalability, security, availability, and maintainability. The volume of errors and issues caused by the lack of comprehensive design indicates that this system's performance issues are caused by not appropriately addressing these technical requirements in the design.

3.4.3 Data

The following additional observations/findings support the summary findings found in subsection 3.4.3 of the Technical Assistance Report:

- There appear to be gaps in the consistent understanding and use of data fields. It appears there were numerous instances where data elements were incorrectly used. The SI data team created a data dictionary (the data dictionary is in ERwin), but it is not clear that there is enough information/access for the SI vendor to code it correctly.
- The SI currently runs approximately 50–60 nightly scripts to clean up the data. It was reported that fixes have been developed for about 95 percent of the data integrity issues, but that only 24 percent of the fixes have been corrected in the production code. Another 70 percent are in Release 1.1; however, even after Release 1.1 is put into production, it will still be necessary to run approximately 20 nightly scripts to clean up the data.
- The following attributes of complete, correct, and feasible data are not completely implemented:
  - **Data Accessibility –** The system does not always allow users access to the data they need, e.g., account locking, inability to login.
  - **Data Comprehensiveness –** The system does not collect all data required for supporting functions, e.g., the time-outs caused by real-time calls for rates. During the assessment, it was not clear that the developers understood how the data should be used and what steps would ensure complete collection of end-user data.

- **Data Consistency –** The data received by the system is not current at a specified point in time and is unable to support its intended business needs.
- **Data Precision –** It is not clear if data attributes and values are defined at the correct level of detail.
- **Data Timeliness –** Data is not being processed in a timely manner and does not support business needs. It is not clear if all data needed to construct a business context was available to the system and processed without delay or if data loads and error corrections were completed in time. This issue was clearly present in the Notice generation process that created bottlenecks (and for which there was no error handling).
- **Data Validity –** The MITRE team found data in the system that was considered valid but did not have correct values. For example, the loading of incorrect plan rates did not trigger any error message.
- **Non-Duplication –** It was not clear if the system recorded data correctly without any duplication for the same entity, relationships, or correlating events.
- **Usability –** The information in the system was not usable for its intended purpose.
- **Completeness** – It was unclear whether the system stored all required data. Completeness implies that a system has a high degree of knowing and storing all required data. This includes having all required data elements, their values, data records, and all facts about the object or event that caused alterations to the data.
- **Relationship Validity –** It was not clear if the system had data that conformed to the associative business rules.
- **Data Accuracy –** It was not clear if the data in the system accurately reflected the expected data or event. The data processing was not complete and fully tested, and the ensuing result was not available to ensure accuracy.

- The addition of the worker portal to the SI vendor's workload required a new look at the data structure and the development of an updated data architecture that could accommodate the additional data needs for this functionality. It is expected that the system will need to migrate to this new data structure within the next few months.

3.4.4 Integration

The following additional observations/findings support the summary findings found in subsection 3.4.4 of the Technical Assistance Report:

- There are numerous issues in the architecture, design, and data that prevent an optimal functional and technical integration. This requires examining a broad range of technical components, including services, data, hardware, network, application, and platform (among others) to complete a thorough review. Based on the evidence presented by the current system in operation, it is clear that integration is currently incomplete and faulty.
- The SI vendor also related to the assessment team that the code developer did not know a business rule was available in the business rules engine for use in computation, so the rule did not get called and the solution was hard coded instead. This demonstrates a classic failure of data and functional integration.

- During conversations with the integrated team, MITRE observed that project teams constructed the system's functions and capabilities in silos. Compounding that problem was the lack of an integration specialist or a fully functioning Chief Architect. These are leading indicators for integration problems. In addition, data integrity issues have arisen due to faulty integration of the independently developed code.
- The assessment also revealed that the SI vendor is implementing verification services within the portal layer and accessing the database through the Hibernate data access package via Spring. By not taking advantage of data abstraction, the SI vendor might have alleviated some integration challenges.
- When determining rates for the various plans, the system is designed to call the issuers for rate information and merge the rate information with the plan design information stored locally. This is done via multiple methods, including XML calls to carrier-developed or hosted services for seven carriers, a locked Excel spreadsheet for one carrier, and Excel spreadsheets for two other carriers. The SI vendor did not have specific performance information, but it was noted that the system was set up with a 20-second time-out. There were occasions when some plan information was not displayed to the customer because of the time-out period. Whether the system experienced a time-out or the time-out was based on a complex series of factors (i.e., not just the data submitted), the results were not consistent and the user could have seen a different number of available plans even when resubmitting the same data. This variability causes a loss of credibility.
- The system is not set up to capture any transactions times, although these can be derived from information in the logs. Since the data was not easily accessible, it was only used when absolutely needed for troubleshooting a specific issue. Since some variability was considered reasonable by the Commonwealth, it is not clear that the SI vendor was conducting performance analysis for this variability. It was noted that although this process seemed to add latency to the new system, the legacy system had functioned adequately using a similar process.
- Best practices for integration testing suggest that a functional lead usually works with SMEs and the Data Architect to simulate appropriate synthetic data. The test data, in association with complex test cases, ensures sufficient test coverage. Automated testing tools or scripts codify the functional complexity before and during continuous testing so that regression testing gains in maturity over time. MITRE's assessment of the Health Connector determined that the lack of such discipline and standardization has caused test data to fail, resulting in incomplete testing. Test data should be developed to allow for testing at various integration levels so that the system can be built incrementally based on fully tested services.
- The system does not currently handle special enrollment periods nor reporting of changes in circumstances (e.g., additions to family, change in family structure).

3.4.5 Infrastructure

The following additional observations/findings support the summary findings found in subsection 3.4.5 of the Technical Assistance Report:

- The system does not currently include any application-based monitoring systems. The SI vendor indicated it was concerned about data access and control issues in handling federal tax information and personally identifiable information and were not keeping full logs due to the potential for an incident. The SI vendor requested further guidance and assistance in determining what use of production data is acceptable. It was noted that this resulted in difficulties identifying the root causes for several errors (BA000 and BE008).
- It appears that the production infrastructure is currently workable (although not perfect) for HIX/IES. There is a lack of tools to perform application monitoring, and there is no systematic review of the automated logs that are kept. Configuring, sizing, and monitoring the platform to perform the functions of the HIX/IES are ongoing activities. Infrastructure in this context refers to such platform tools as Oracle's Fusion middleware and databases, network and storage platforms, hardware, and operating systems, including configurations in the data centers.
- There is a lack of systematic application monitoring and correlation of incidents to the events recorded through review of logs generated by the platform tools—i.e., a root cause analysis. This needs to be rectified through the data center operational processes. These processes will help the SI vendor and Commonwealth fully understand whether or not the infrastructure partly causes the current system issues.

3.4.6 Key Technical Areas

The following subsection provides additional information regarding a number of key technical areas. The Technical Assistance Report did not address these areas in detail.

3.4.6.1 Account Creation

The system uses Oracle Identity Manager (OIM) and Oracle Access Manager (OAM) products for Account Creation and management. There were many issues with Account Creation, and the Commonwealth set up a SWAT team to investigate and resolve these problems. Some issues were traced to the system's failure to assign a role to an account once created. This was mitigated through the use of a periodically run post-account-creation script that assigns the lowest level of roles to all accounts that do not have the role set. The main reason for this may have been issues with the sizing of the environments, which may have restricted flows. The SI vendor increased the number of servers supporting the portal and access management systems and has decreased the incidence of account creation failure. Further changes are under consideration that transform the Account Creation function from a two-step process—one to create the account and a second to set the role—into a single-step process. This is one of the areas where effective error trapping and user-friendly explanations have not been employed.

In addition, MITRE noted that the password complexity requirements displayed did not match the actual password complexity requirements established for the system. Users could meet the criteria displayed to them and still be unable to successfully create an account. There appears to be a correlation between the server size increase and the decreased incidents of errors, and it seems that some progress is being made The absence of effective performance testing does not provide any confidence that the system issues have been resolved. The SWAT team identified several other methods for reducing the loading on the OIM/OAM products, but it is not clear what benefits these will produce.

3.4.6.2 Portal Layer (Application Data Entry)

The MITRE team looked at several portal screens and found that errors were not being trapped and recast in user-friendly formats, resulting in the user's inability to continue. At some points, the system did not let users proceed to the next page without providing some indication of the issue. Although the basic portal screens do not appear to be a source of problems, the connections and actions the system takes to transition between screens are frequently problematic. Because change in status functionality has not yet been completed, members cannot currently report changes in their family situation such as income changes, pregnancies, births, or deaths using the system.

3.4.6.3 Eligibility

The current status of Program Determination is that the system can process individuals who are not applying for financial assistance. Joint testing has been performed on most of the simple individual applicant scenarios where the individual is requesting subsidized assistance. A few changes are required to allow temporary categorization of a small percentage of this population, subject to a later recheck or recharacterization. Minimal testing has been done on "couple" and "other family structure" applications. Although the rules engine testing done using the SACT seems to indicate that the rules engine is performing well, there is not enough information on whether it will handle the more complex groups. MassHealth indicated it has approximately 3,000 test cases for the "couple" and "other family structure" Medicaid population that still need to be checked. It seems likely that CCA will also have a significant number of cases that will need to be checked.

Once eligibility has been determined, the generation of Notices is initiated. There appear to be problems with the Notice generation, although there is significant progress in this area. Since this portion of the system is currently executed only as a part of the end-to-end testing (and issues are occurring earlier in the process), it appears that considerable testing is still necessary.

3.4.6.4 Shopping and Enrollment

MITRE also assessed the Shopping functionality. The system has been processing individuals not requesting financial assistance since October 1, 2013, but the provider search function is not working. When executing a provider search, the user sees fields without headers, an incomplete list of languages, an incorrect list of specialties, and other issues. When the system develops the list of available plans and their rates, it makes automated calls to carriers for rate information. This process has a time-out in place, and if a time-out occurs, the user may see different numbers of plans returned for two executions of the same search. The layout of the displays for the prices and the individual versus family cost characteristics indicates that further work on the user interface is needed.

4. Detailed Recommendations

Section 4 of the Technical Assistance Report submits a set of ten (10) near-term (30–45 days) and three longer-term (beyond 45 days) recommendations.

This section provides additional detail about the near-term recommendations based on the findings found in Section 3 of the Technical Assistance Report and this supplement. The recommendations are organized by the same four main topic areas used in Section 3 and the broad categories. Neither the order of the categories nor the content within the categories implies a suggested priority.

4.1 Project Structure, Roles, and Responsibilities

This subsection provides additional detail for recommendations one through three from the Technical Assistance Report:

- Establish one full-time senior government executive accountable for the success of ACA program planning and execution for the Commonwealth.
- Draft a governance model that defines clear roles and responsibilities, and stand up governance bodies. (***Note***: Subsection 4.3 addresses OIT governance.)
- Develop an integrated program structure that includes leaders for the Health Connector, MassHealth, and UMMS and a small program management office (PMO). Include a specific technical structure as part of the program. The structure should include a project technical manager and best practice technical resources [dedicated SMEs, Chief Architect, Chief Engineer, Data Architect, QA/Quality Control (QC), etc.]

MITRE respectfully submits the following recommendations in this area:

- The program should consider reorganizing with stronger lines of authority, accountability, and control over the resources necessary to implement the new system.
- The Commonwealth should develop an integrated program structure led by a full-time senior government executive empowered to establish priorities, make decisions, and manage scope. The current lines of ownership for development of this system hamper the ability to implement the system. The Health Connector and MassHealth each have their own priorities; UMMS is the contract manager, negotiating and administering changes driven by others. A single person with complete authority and responsibility to oversee and manage the program would facilitate a more cohesive program.
- The SMEs must report directly to the project manager. Currently, project staff relies on other organizations to provide SMEs. Those SMEs focus on their parent organization's needs rather than the development of the HIX/IES. The program manager should manage this relationship.
- The project needs the following programmatic and technical support:
  - A Project Manager empowered to enforce project and systems engineering best practices and delegate responsibility to team leaders
  - A Business Analyst to provide leadership and act on behalf of the business owner

- A Technical Architecture team to provide technical recommendations to the Project Manager and approval boards (Change Control/Configuration Management)
- A Technical Integration Manager to ensure all parts of the system work correctly and as designed
- Dedicated technical resources, including Quality Assurance/Quality Control, SME support, Chief Engineer, Data Architect, and process leadership

4.2 Project Planning and Execution

The Health Connector currently has a small staff providing management and technical oversight for the project. Given the project's complexities, additional oversight is recommended. On the systems integrator side, the team needs to focus on the processes outlined in Table 1. Each process should have a task lead and team staffed to meet the project's needs. These processes and their rigorous execution are critical to the project's success.

Table 1. Process Descriptions

Process	Process Function
Project Management	Maintain and enforce the Project Management Plan. Periodically report on project progress and performance in the areas of schedule, scope, and budget.
Scope Management	Maintain scope definition and identify risks when scope exceeds available resources.
Schedule Development and Management	Develop, maintain, and manage to the integrated project schedule. Manage work breakdown structure and work alignment to ensure appropriate resources are available when needed.
Quality Assurance	Ensure that appropriate quality processes are built with sufficient rigor and the team is applying standards and policies.
Quality Assurance Manager	Ensure that the test team is applying sufficient testing rigor and the Quality Manager has the authority to accept or reject the build/delivery.
Communications Management	Develop and use a detailed communications plan and processes to ensure information is properly disseminated to the project team and others who have a need to know.
Risk Management	Ensure that the program actively tracks and manages risks and issues, and periodically produces management risk reports.
Integrated Change Control	Ensure that requirements baselines are current, designs are updated, and changes are traced and applied to appropriate technical components.
Integration Manager	Ensure that the project actively maintains the technical blueprints, and ensure integration among system components.
Business Architect	Ensure that the implementation has the capability to adapt to an agreed-upon set of requirements and harmonize the requirements with other functional areas. The architectural staff may include additional resources that understand the Commonwealth's business processes and can work with the development team to ensure a smooth implementation.

Process	Process Function
Process Management	Ensure that process expertise is available to oversee and monitor the implementation and execution of program/project management and systems engineering processes. Consider creating an integrated Process Assets Library, including templates, tools, procedures, and Responsible / Accountable / Consulted / Informed (RACI) documents that everyone can follow.
Data Architect	Ensure that the appropriate data structures, operators, and technologies are applied in achieving the integrity of data creation, transmission, transformation, storage, and security.

4.3 IT Governance / Software Life-Cycle Process

This subsection addresses processes associated with recommendations six through ten from the Technical Assistance Report:

- Place baseline requirements under configuration control.
- Identify and prioritize program risks.
- Draft and execute a twelve (12)-week Program Plan.
- Create an overall, high-level program road map.
- Produce and maintain an Integrated Master Schedule.

Governance is severely lacking and must be strengthened to improve the quality of the HIX/IES development. Governance oversight throughout the life cycle has been cut short to expedite development and implementation, but this has resulted in a loss of quality and marginal system performance, at best.

For example, it appears that there is no formal organized UAT for Release 1.1, and project decision-makers are making decisions based on incomplete or missing information, including the quality of Release 1.1. Project management does not have complete data showing the breadth of regression testing nor the completeness of testing on the data integrity issues plaguing the system. Given the need for immediate improvement, it is necessary to implement sound governance practices in the areas listed in Table 2 as soon as possible.

Table 2. Governance Areas

Governance Area	Impact of Governance
Requirements Analysis and Management	• Ensures that requirements are always current and changes are tracked • Enforces traceability to the design with integrated review against the requirements • Ensures that the SI vendor and testers have access to current requirements • Ensures that appropriate management reviews and sign-offs occur throughout the process
Coding Standards	• Ensures that coding standards are defined, updated, and enforced

Governance Area	Impact of Governance
Change Management	• Ensures that change management processes are integrated with the risk management process so that risks are evaluated and considered when assessing changes
Build / Release Management	• Ensures that the process for managing builds is improved, and testing and complete sign-off occurs prior to transition • Ensures that there is enough time to develop test cases and conduct reasonable UAT • Ensures that there is a process where the release authority (configuration control board) is provided adequate data to make decisions • Ensures assembly and testing of more intermediate builds to gain earlier identification of issues and concerns to enable proactive problem handling
Design / Implementation	• Ensures that the design/design review process is reinstituted so better integrate the development efforts to meet the functionality required • Ensures that the design includes performance characteristics for components • Ensures reviews of the translation of business requirements into technical components with participation by Commonwealth SMEs • Ensures that the establishment of a project-level architecture review board—comprising representation from the application, data, network, security, and middleware domains—that meets frequently • Ensures documentation of architectural decisions • Ensures, at a minimum, the adoption of an Information Technology Infrastructure Library (ITIL) Standards and Service Management for the Enterprise, if that function does not currently exist • Ensures that the project move to a function-oriented rather than a production-line approach to managing capability • Ensures that adequate procedures are followed to better deliver full capability in release
Testing	• Enables restructuring of the testing approach to incorporate more intermediate testing • Ensures that adequate verification processes promote verification and test case development early enough in the process to flesh out test data and test cases before must be executed • Meets the need to incorporate multiple layers of testing to verify integration is progressing smoothly and has appropriate processes and resources to ensure system performance is meeting the business functionality • Ensures that a restructured testing program incorporates more automated testing • Ensures that the testing process is integrated into the build release process • Ensures that test scripts are added to the test case library to verify that defects have been resolved • Ensures that the regression testing plans are agreed to by both the CCA and the SI vendor

Governance Area	Impact of Governance
Configuration Management	• Facilitates orderly management of system information and system changes
Architectural Design	• Provides greater visibility into how the design implements the business requirements
Integration	• Ensures that system components work together and function as required
Verification Processes	• Ensures that thorough testing of both the normal and unusual paths are completed and signed off prior to entering production

4.4 Technical Elements

This subsection provides additional detail for recommendations four and five from the Technical Assistance Report:

- Decide on next steps relative to the current system based on three primary options, as follows:
  - **Option #1:** Engage a new SI vendor and start over by building a new system.
  - **Option #2:** Engage a new SI vendor and continue incrementally improving the current system.
    - ***Note*:** Options 1 and 2 are predicated on creating a transition plan and transferring responsibilities from the current SI vendor to one that has proven past performance success in Health Insurance Marketplace implementations constructed on the Commonwealth's preferred platform (Oracle).
  - **Option #3:** Segment the current system into distinct technical segments and then re-engineer each segment as required to complete the system.
    - ***Note:*** Option 3 could be accomplished using a combination of the current SI vendor and additional vendors with expertise to complete the re-engineering.
- Execute the contract actions required to implement the option chosen.

The development and implementation of the HIX/IES system is not going as planned. The system could not be rolled out by the October 1, 2013 deadline and is still not ready for implementation. Subsection 3.4 addressed several overarching areas of concern that have prevented a successful implementation, including overall architecture, software design, data, integration, and infrastructure. These areas are so fundamental that the Commonwealth might consider completely redoing the design and development. This subsection includes three options that should be considered in proceeding with this program.

4.4.1 Start Over / Rebuild the Entire System (Option #1)

The Commonwealth could consider restarting the HIX/IES development and implementation with a new SI vendor. Although parts of the system could be reused, it might be more efficient to hire another vendor or use another state's system that already has the basics in working order, given the challenges in repairing bugs and adjusting design late in the development life cycle for

an already developed system. It is likely that the Commonwealth will need to tailor any system in order to handle specific programs. MITRE respectfully submits that the Commonwealth consider the following:

- Create a transition plan and transfer responsibilities from CGI to a vendor that has the IT skills and subject matter depth in constructing complex, enterprise-class applications on the Commonwealth's preferred Oracle Platform. This approach includes:
  - Re-engineering the current application architecture with a newly acquired SI vendor
  - Re-engineering the middleware components to establish core technical services
  - Refactoring and transitioning the rules engine and the workflow into the new architectural framework

4.4.2 Continue Working to Incrementally Improve the System (Option #2)

This option implies retaining the current SI vendor, but changing the SI vendor's processes and methodology. This option emphasizes governance and oversight, with much closer inspection of the SI vendor's work by the Commonwealth. To follow this approach, MITRE recommends that the Commonwealth:

- Restructure both the SI vendor and Commonwealth teams to provide focused expertise to jointly complete functionality for the system
- Ensure that there are user-friendly error messages where appropriate
- Minimize any new work scope to the current SI vendor's workload, and carefully evaluate any proposed new expansion of work scope
- Conduct extensive testing
- Move work that can be easily separated, such as the Small Business Health Options Program (SHOP), to alternate vendors

4.4.3 Segment the System into Technical Areas and then Re-engineer / Improve Different Areas in Isolation (Option #3)

This option could be used if a new SI vendor is engaged, or if the Commonwealth elects to continue with the current vendor. A new SI vendor could use the existing segments as a basis for either creating a new design, development, and implementation, or for improving the existing implementation. The new vendor could concentrate its design and development efforts on the assigned segments individually, with parallel teams, and integrate all segments when they are ready.

The segmentation effort includes:

- Leveraging some SOA break points to separate out the following integrated activities:
  - Application Intake
  - Program Determination
  - Notices Generation

  - Shopping
  - Billing
  - Identity and Access Management
- Rebuilding segments using the following methodology:
  - Each segment has its own development team. This development team is responsible for testing all components of the segment prior to its integration with other segments.
  - Rebuild segments in a standalone environment
  - Consider acquiring a current implementation of a like segment from another state-based marketplace state
  - Ensure that each segment is capable of decoupling and recoupling with the rest of the software components
  - Engage an integration specialist whose sole responsibility is to reestablish the connection among segments
  - Engage a data architect to trace data flows and evaluate and recommend specific improvement to the data structures
  - Conduct extensive testing at all levels and verify data integrity after each test on all systems/components to be retained

4.4.4 Considerations for Selecting an Option

The best option is a function of the Commonwealth's priority. If the priority is to have an exceptional website, Option 1 is likely the best choice (and in all likelihood the most costly and disruptive option). If the goal is to provide a workable system in the near term, Option 2 would be preferable, provided the SI vendor can implement the recommendations. Option 3 could be used by either a new contractor or by current SI vendor. The new SI vendor would have a good start on a new system, while the current SI vendor could divide work and make faster progress.

Given the challenges with the current system, it appears that a multifaceted approach might be warranted. This would include both triaging the current system while working to upgrade key system parts. In the short term, a team of experts should triage the current portal layer. This team should be empowered to make corrections to capture system errors, convert them to user-appropriate language, and improve nomenclature and labelling to ensure that users can navigate through the process. At the same time, the Commonwealth should set up separate teams to continue the joint testing on eligibility determination and notices. The system should be designed to conduct testing in these areas in parallel, with automated testing of data sets that have been thoroughly planned to check both the normal and unusual paths through the processes.

The Commonwealth should consult with CMS on funding as well as the agency's guidance on or approval of these recommendations.

Acronyms

ACA	Affordable Care Act
APTC	Advanced Premium Tax Credit
CAMH	CMS Alliance to Modernize Healthcare
CCA	Commonwealth Connector Authority
CCIIO	Center for Consumer Information and Insurance Oversight
CMCS	Center for Medicaid and Children's Services
CMMI	Capability Maturity Model Integration
CMS	Centers for Medicare & Medicaid Services
EOHHS	Massachusetts Executive Office of Health and Human Services
FFRDC	Federally Funded Research and Development Center
HIX/IES	Health Insurance Exchange/Integrated Eligibility System
IAPD	Implementation Advance Planning Document
IT	Information Technology
IV&V	Independent Verification and Validation
MA	Massachusetts
MAGI	Modified Adjusted Gross Income
MMIS	Medicaid Management Information System
OAM	Oracle Access Manager
OIM	Oracle Identity Manager
OPA	Oracle Policy Automation
PPACA	Patient Protection and Affordable Care Act
RACI	Responsible, Accountable, Consulted, Informed
SACT	Semi-Automated Contingency Tool
SHOP	Small Business Health Options Program
SI	Systems Integrator
SME	Subject Matter Expert
SOA	Service-Oriented Architecture

UAT	User Acceptance Testing
UMMS	University of Massachusetts Medical School
XML	Extensible Markup Language