ElyonStrategies

Cal-Access Replacement System Independent Assessment Advisory Services

Project Assessment

Title	CARS Project Assessment		
Document #:			
Version:	DRAFT	**DATE:**	12/11/2021
Author(s):	SCOTT OHLUND, KEITH MOREHOUSE, WILLIAM ROETZHEIM		
Customer:	California Secretary of State (SOS)		
Contract:	21C60064		
Project:	Cal-Access Replacement System Independent Assessment Advisory Services		
Deliverable ID:	C.2 THROUGH C.14 PLUS E.1		

VERSION HISTORY

Version	Author	Date	Changes
DED	WHR	9/8/2021	Original DED
DED-Update	WHR	11/3/2021	Updated DED based on WOA 2 findings
Draft	Multiple	12/1/2021	Original Draft
Draft	Multiple	12/4/2021	Added in Chapter 6
Final	Multiple	12/11/2021	Revised based on State comments.

Approvals

Role	Name	Signature	Date
Author	Keith Morehouse		12/11/2021
Author	Scott Ohlund		12/11/2021
Author	William Roetzheim		12/11/2021
QM	Karen Morphy		12/11/2021
CEO	Carl Engel		12/11/2021

Table of Contents

Contents

List of Tables

List of Figures

1 Executive Summary

The California Automated Lobbyist and Campaign Contribution and Expenditure Search System (CAL-ACCESS) is the public's window into California's campaign disclosure and lobbying financial activity, providing financial information supplied by state candidates, donors, lobbyists, lobbyist employers, and others. CAL-ACCESS is the mission-critical legacy system for the Secretary of State (SOS)'s administration of the Campaign Finance and Lobbying disclosure program. The CAL-ACCESS Replacement System (CARS) project is tasked with implementing a new system that replaces the legacy CAL-ACCESS solution with a modern technology-based, data-driven system. This system will allow campaign and lobbying entities to meet the filing requirements of the Political Reform Act (PRA) more efficiently, improve data quality, expand public access to data, allow for system modifications and improvements to respond to statutory and regulatory changes, allow other system modifications to improve filer efficiency and public access to data, and improve the ability of the SOS, the Fair Political Practices Commission (FPPC) and the Franchise Tax Board (FTB) to fulfill mandated duties.

Elyon Strategies was hired to assess the current health of the CARS project holistically and objectively. As part of this assessment, Elyon is working closely with the SOS and the CARS team, including its various vendors, to effectively evaluate the business, technical infrastructure, and project management practices to provide a corrective action plan and roadmap. The results of this assessment will guide the SOS in general, and the CARS Project Sponsor in particular, with a path forward to develop a remediation plan to drive towards achieving successful completion, implementation, and delivery of the CARS system with the goal of meeting or exceeding Political Reform Division (PRD) and external stakeholder business needs, fulfilling legislative and statutory requirements, and functioning consistent with the legislative intent stated in Government Code section 84601 as well as other provisions of the Political Reform Act. The intent is for the SOS to use this document as a form of systematic lessons learned, assessing the current (as-is) state of the CARS project, and pointing the way forward to a to-be project state.

Chapter 2 provides background information about the project and our assessment approach; Chapter 3 provides a summary of our findings and recommendations; Chapter 4 provides specific results of our assessment in the technical aspects of the project; Chapter 5 provides specific results in areas of technical management; Chapter 6 provides results in aspects of project governance and oversight; and the appendices include supporting material, including our detailed assessment criteria and observations.

CARS business requirements are characterized by complex business rules that are unique to this application and subject to change over time; complex workflow and form-flow requirements that are integrated with the business requirements; significant descriptive, relationship, and financial data that needs to populate to forms and reports; requirements to accept various degrees of "dirty" data for subsequent correction; and extensive version control and redlining at the field level. The application is like a taxation system with requirements for form or wizard-based data entry in accordance with

complex requirements, plus compliance reviews/audits. A key requirement of the external community portal is the ability to query, tabulate, and compare data across multiple years.

The project encountered difficulties with the selected implementation vendor (Perspecta), and roughly four years into the effort switched to a new primary implementation vendor, Outreach Solutions as a Service (OSaaS). At the time of that OSaaS contract award, the legislatively mandated deadline for completion of CARS was approximately eight months away. In June of 2021, four-months after the target completion date, the project was paused pending an assessment and development of a go-forward strategy. The CARS project timeline is shown in Table 1. Perspecta was under contract to deliver CARS from February 1, 2016, through June 30, 2020. Following termination of that contract, OSaaS was under contract to deliver CARS starting on July 1, 2020, through February 28, 2021.

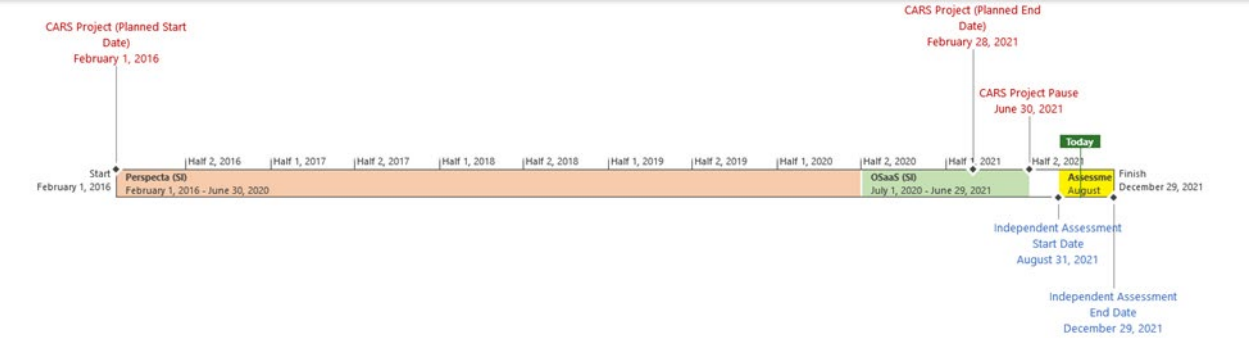

Figure 1: CARS project timeline.

Projects whose project parameters exceed the norm by 50% are characterized by heroic efforts, long hours, burnout, and in the end, failure. The task assigned to the CARS project team overall, of doing several years-worth of development work in eight months, was quite simply impossible.

One characteristic of these projects that are well outside the norm is that time pressure means the vital foundational and architectural work is rushed or skipped altogether. The team moves forward to begin the development phase, without a clear understanding of what needs to be built, and without an optimum and supportable underlying architectural structure. In virtually every case, the project begins to quickly build something, but they are building the wrong thing, and they are building it the wrong way. CARS was no exception to the rule.

Unfortunately, when an Information Technology (IT) project has these foundational problems, much of the software that has been developed has limited use. Even when it can be patched and extended to support the updated architecture, it will suffer from problems in areas including reliability, maintainability, security, and performance. For much of the developed application, the total cost of ownership to rework or repurpose the software correctly is typically prohibitive for many reasons (e.g., cost).

During our initial 30-day assessment, we conducted a thorough review of the current project state and concluded that:

- While it would be possible to build CARS using Salesforce as part of the solution, Salesforce does not offer sufficient value added in the form of reusable functionality to justify requiring that the solution be based on Salesforce.

- The existing system is flawed at the architecture, data structure, middle-tier, and user presentation layers. The architecture, data structures, and middle tier are not correctly optimized for the necessary business processes, workflows, and data structures. The user presentation layer takes a purely form-centric view of the world, as opposed to an underlying data-centric view. Those flaws are fundamental and recasting the project will be more cost effective than to continue development. Anything less will likely result in an unreliable system with significant functional deficiencies that is expensive and difficult to maintain.

- The most cost effective, and lowest risk, approach to implementing CARS is to salvage reusable components from the work to date, then conduct a new acquisition for a CARS restart using an alternate architectural approach. With this approach, CARS can be implemented with a high degree of confidence, and the resultant system will save significant funds over time through improved functionality, maintainability, and reliability.

- The project should move to a firm-fixed price deliverable-based contract model, and a competitive acquisition should be used for the contract restart.

In conducting our CARS assessment, we used our enterprise Maturity Readiness Indicator (eMRI) project assessment modeling tool. The assessment was conducted for the project as a whole, so including both SOS and contractor activities. The models use a weighted multi-variate assessment approach to arrive at an overall project assessment of 1 to 5, where 1 is Very Low, 3 is typical or average, and 5 is Very High. Overall, the assessed score for CARS was 2.34. This score was derived from a detailed analysis across the following eleven process categories:

1. Data Conversion and Migration.
2. Release Management.
3. Testing.
4. Requirement Definition and Management.
5. Project Schedule Management.
6. Risk Management.
7. Contract Management and Vendor Negotiations.
8. Communication Management.
9. Governance and Sponsorship.
10. Organizational Change Management.
11. Quality Management.

In developing the specifications for the CARS restart, consideration should be given to a phasing strategy that will deliver some intermediate value-added capabilities as the project is completed. These would be items that offer value to internal or external stakeholders; that could be delivered with no or minimal incremental impact to overall project cost; that are feasible from a business process perspective; and that are technically feasible. For example, based on a workshop conducting with PRD personnel, we believe that the following intermediate deliveries might fit these criteria:

- Early creation of the CARS database. This database could then operate in parallel with the current CAL-ACCESS Oracle Database Management System (DBMS). The database should be structured to allow but tag dirty data. Among other benefits, this would allow the CAL-ACCESS data to be easily migrated to the new database. Following the migration, a refresh capability should be implemented from CAL-ACCESS to the new DBMS.
- Implementation of a public portal to report against the new DBMS.
- Implementation of code to receive electronic files from external vendors using the current file format. A lookup table would then be used to split the data, with some form types loading into CAL-ACCESS and some form types loading into the new DBMS.
- Installation of the new business rules engine and forms engine. Using those capabilities, new and replacement forms could be created to feed data to the new DBMS.
- Following these early milestone deliveries, the remainder of the CARS capability could be implemented.

In addition to our phased implementation recommendations above, our specific recommendations for the CARS restart going forward are shown in Table 1, with the indicated priority for each in terms of ensuring that the CARS restart is successful. Each of these recommendations is expanded upon in Chapters 4, 5, and 6. The priorities have the following meanings:

- Critical: These factors were identified as having the highest impact on the success of the CARS restart.

- Highly Important: These factors are important to the success of the project, and failure to implement these may put the project success at risk.

- Important: Implementing these factors will have a significant impact on some combination of risk, cost, or schedule.

Table 1: Elyon go-forward recommendations.

Critical Recommendations	
Area	**Recommendation**
Governance and Sponsorship	A Project Charter should be created and approved. A Project Sponsor should be identified and be given the responsibility and authority

	necessary for project success, including the final decision authority with respect to moving forward beyond each gate review.
Organizational Change Management	After selection of the technology solution as a result of the acquisition, a skill gap analysis should be conducted for all stakeholders, internal and external, and including business, executive, technical, and support stakeholders. This gap analysis should be used as a primary input to the OCM and training plans.
Contract Management	The CARS recompete acquisition should be structured as a competitive firm fixed price deliverable-based contract, thereby allocating the risk to the party best able to manage that risk (the vendor). Work where the level of effort is largely under the control of the SOS, specifically in the areas of training, transition, maintenance, and operation may best be structured using fixed labor hour bids.
Requirement Definition and Management.	CARS should: • Review all requirement related material, identify the most correct version of the most useful artifacts, and place that material under configuration and version control. A copy should be included in the bidder's library. • Clarify the role of automation with respect to the requirements (especially the business rules). In many cases it is currently unclear if a given requirement will be fully automated, partially automated, or remain a manual or external process. • Clearly differentiate between as-is and to-be processes, where as-is process descriptions may be useful for background information, while the to-be processes are the actual requirements to be verified as part of the new system acceptance.
Requirement Definition and Management.	CARS should ensure that non-functional requirements impacting on the technical architecture are called out going forward. Specifically: • The requirement to accept but tag dirty data for later reporting and clean-up. This may include differentiating between errors and warnings in the data tags, and potentially include a severity level. • The requirement to track version changes to the field level for most data. • The requirement to easily change business rules and workflows, using data configuration rather than programming, when possible.
Testing.	CARS should require a Test Readiness Review (TRR) prior to commencement of System Integration Testing (SIT) and User Acceptance Testing (UAT), and a Production Readiness Review (PRR) prior to release to production. If the phased deployment approach is implemented, then a TRR and PRR would be conducted for each of the phased releases. These reviews should be gate reviews.
Schedule Management.	Formal estimation techniques should be used when estimating resources and schedule.
Schedule Management.	Gate-review milestones should be included in the CARS project schedule. These gate reviews should occur no less often than every six-months during the life of the project. They should include a Software Requirement Review (SRR), Detailed Design Review (DDR), TRR, PRR, and a Post Implementation Evaluation Review (PIER).

Highly Important Recommendations	
Area	**Recommendation**
Data Conversion and Migration.	CARS should require the implementation vendor to convert the CAL-ACCESS data early in the Software Development Lifecycle (SDLC) for the CARS restart. If the phasing strategy recommended earlier is adopted, the phasing approach will accomplish this automatically.
Data Conversion and Migration.	CARS should require the implementation vendor to develop a data conversion plan that also addresses data validation and ensure that the data validation approach provides the necessary auditable assurance of data accuracy and completeness.
Testing.	CARS should begin assembling and preparing test cases as soon as possible, with the objective of having most test cases complete by early during unit testing and all test cases complete prior to the start of SIT. Test cases should address normal, exception, and boundary conditions.
Requirement Management.	We recommend that a new requirement management process be implemented as part of the requirement review process recommended above. This requirement management process should include, among other things, requirement related governance.
Communication Management	A project health dashboard and vendor performance balanced scorecard should be developed and updated on a regular basis.
Quality Management	All project artifacts, including hardware, software, code, engineering artifacts (documents), process documentation, and requirements should be placed under configuration control, with one approved and most current version.
Quality Management	An IV&V vendor should be brought on-board to focus on verification and validation of contractual products, including internal and external consistency, correctness, and fit-for-purpose.
Quality Management	Project processes should be documented, optimized, and approved.
Important Recommendations	
Area	**Recommendation**
Data Conversion and Migration.	The implementation vendor should be required to create and obtain approval of a security plan that includes data security during the conversion process.
Release Management.	The implementation vendor should be required to create and obtain approval of a Release Management Plan for each software release, including phased releases, that describes the release's activities, schedule, resources, roles, responsibilities, risks, and roll-back criteria and strategy.
Requirement Management.	We believe that there is an opportunity during the requirement refresh to go even farther than the existing use-case scenarios in terms of potential opportunities for improving the way the PRD business processes work. For example, we believe that by allowing the system to accept dirty data that is automatically tagged with error or warning conditions, automated processes may be put in place to support the cleanup of that data through outreach to the data submitter, through automated processes, or through workflows for PRD analysts.
Schedule Management.	Resource leveling should be used to ensure that project resources are not overloaded.
Risk Management.	Risk managers should actively seek and accept risk related input from all stakeholders, both internal and external. Risk input should be utilized no matter what approach is used to communicate the risks (e.g., email, spreadsheets, phone, verbally).

Communication Management	A project website should be established and maintained to communicate project status and information to internal and external stakeholders.
Quality Management	A quality assurance group should ensure that project processes are correctly followed.

In our final deliverable under this contract (Work Order Authorization (WOA) #4), we will provide a CARS Roadmap Forward that will provide the recommended budget, schedule, and resources necessary to ensure a successful project. This document will contain the CARS restart estimate to complete, including estimation boundaries, inclusions, and exclusions; application scope; labor costs; project specific estimation adjustments; other direct charge assumptions; and the benchmark driven estimates for project schedule, resources, staffing, artifacts, and on-going maintenance and operations costs. The Roadmap will also include an estimation related risk analysis.

2 Introduction and Approach

2.1 Project Background.

The California Automated Lobbyist and Campaign Contribution and Expenditure Search System (CAL-ACCESS) is the public's window into California's campaign disclosure and lobbying financial activity, providing financial information supplied by state candidates, donors, lobbyists, lobbyist employers, and others. CAL-ACCESS is the mission-critical legacy system for the Secretary of State (SOS)'s administration of the Campaign Finance and Lobbying disclosure program. The CAL-ACCESS Replacement System (CARS) project is tasked with implementing a new system that replaces the legacy CAL-ACCESS solution with a modern technology-based, data-driven system. This system will allow campaign and lobbying entities to meet the filing requirements of the Political Reform Act (PRA) more efficiently, improve data quality, expand public access to data, allow for system modifications and improvements to respond to statutory and regulatory changes, allow other system modifications to improve filer efficiency and public access to data, and improve the ability of the SOS, the Fair Political Practices Commission (FPPC) and the Franchise Tax Board (FTB) to fulfill mandated duties.

Elyon Strategies was hired to assess the current health of the CARS project holistically and objectively. As part of this assessment, Elyon is working closely with the SOS and the CARS team, including its various vendors, to effectively evaluate the business, technical infrastructure, and project management practices to provide a corrective action plan and roadmap. The results of this assessment will guide the SOS in general, and the CARS Project Sponsor in particular, with a path forward to develop a remediation plan to drive towards achieving successful completion, implementation, and delivery of the CARS system with the goal of meeting or exceeding Political Reform Division (PRD) and external stakeholder business needs, fulfilling legislative and statutory requirements, and functioning consistent with the legislative intent stated in Government Code section 84601 as well as other provisions of the Political Reform Act. The intent is for the SOS to use this document as a form of systematic lessons learned, assessing the current (as-is) state of the CARS project, and pointing the way forward to a to-be project state.

Chapter 2 provides background information about the project and our assessment approach; Chapter 3 provides a summary of our findings and recommendations; Chapter 4 provides specific results of our assessment in the technical aspects of the project; Chapter 5 provides specific results in areas of technical management; Chapter 6 provides results in aspects of project governance and oversight; and the appendices include supporting material, including our detailed assessment criteria and observations.

2.2 Approach.

Elyon Strategies is a management consulting and professional services company, providing an integrated service catalog in strategy, architecture, portfolio management,

advisory, assessment and improvement services to achieve complex transformation. Because our focus is on providing independent project oversight support to government agencies, we tend to be involved primarily in the largest and most complex projects. Those are the projects where the need for project oversight is recognized and the budget for oversight is available. Here in California, we have provided this service for many of the State's largest and most complex projects, including:

- California Department of Social Services (CDSS) Child Welfare System (CWS)/Case Management System (CMS) project (a child-welfare case management system).
- Franchise Tax Board (FTB) Enterprise Data to Revenue (EDR) and EDR2 projects (both are taxation systems).
- California Department of Tax and Fee Administration (CDTFA) Centralized Revenue Opportunity System (CROS) project (another taxation system).
- California Healthcare Eligibility, Enrollment, and Retention System (CalHEERS) (a health insurance portal).
- CDSS Statewide Automated Welfare System (SAWS), C-IV, Los Angeles Eligibility, Automated Determination, Evaluation and Reporting (LEADER), and LEADER-Replacement projects (all welfare systems).
- CDSS CWS- California Automated Response and Engagement System (CARES) project.
- California Health and Human Services (CHHS) California Child Support Automation System (CCSAS) project (a child support case management system).

We have also provided this independent advisory / assessment service[1] for large government projects in the states of Washington, Oregon, Colorado, Texas, and Florida; as well as for Federal government agencies including the Department of the Interior, the Department of Agriculture, the Department of Veterans Affairs, the Department of Homeland Security, the Department of Energy, the Federal Aviation Agency, and the General Services Administration. We have provided platform assessments for the various state government departments seeking to modernize legacy applications and determining comparative fit for platforms including Salesforce, Microsoft Dynamics, and Pegasystems. Elyon's consultants have provided applications utilizing these platforms as well, giving our team full Solution Development Life Cycle experience from strategy to implementation and positioning them well to understand both the complexities and benefits of modernization efforts that utilize a platform-based solution.

In performing this work, we model the project using our enterprise Maturity Readiness Index (eMRI) and ExcelerPlan tools. eMRI is an enterprise project process assessment tool, assessing the project process capabilities versus industry best practices, value weighted to the specific processes required for success on each project. eMRI project modeling is in terms of Key Process Areas (KPAs). ExcelerPlan is a benchmark driven

[1] Based on the agency involved, these contracts are sometimes termed Independent Advisory services, IV&V, Project Oversight, Management Performance Audits, Project Management, or Acquisition Support.

system dynamic modelling framework that uses benchmark data to create a model of project success, also tailored to this project. ExcelerPlan project modelling is in terms of High-Level Objects (HLO)s and Function Point Equivalents (FPE)s, which are industry standard ways to define application scope; plus, Other Direct Charges (ODC)s, including infrastructure and licensing; Maintenance and Operations (M&O) support requirements; and project characteristics that impact efficiency. ExcelerPlan's models are based on data from over 40,000 projects.

To date we have conducted forty-nine (49) stakeholder interviews/meetings, reviewed 15,880 documents from a high-level perspective, and identified 1,198 of those documents that are relevant to our analysis. In addition, we reviewed the current Salesforce code in the DevOps system. The purpose of this work was to fully understand the CARS scope and current implementation. The documents that were reviewed were the versions in the SOS SharePoint site, which we believe are the latest version of each document.

In developing our list of priority items for consideration by the CARS project, our criteria were:

- Ensuring that the CARS restart will result in a high quality, maintainable system that meets the objectives of the various internal and external stakeholders.

- Maximizing value to the State of California by ensuring that previous work is used to the greatest extent possible and that the path forward is the most economical one in-line with the above quality related objectives.

- Minimizing the go-forward risk. Our objective is to ensure that the proposed approach is achievable with a high degree of confidence.

- And finally, to the extent possible within the above considerations, we sought opportunities to use a phasing approach to deliver incremental value to CARS users (internal and external).

2.3 Assumptions and Constraints.

Our analysis is based on the following assumptions:

- The state seeks an optimal go-forward strategy. In accordance with Generally Accepted Accounting Principles (GAAP), sunk costs are ignored in performing the financial portion of this analysis[2]. This is because those sunk costs will be the same under all potential scenarios going forward.

[2] For a good discussion of this topic, see: Sunk Cost - Why You Should Ignore Them (the Sunk Cost Fallacy) (corporatefinanceinstitute.com).

- The SOS will go through a PAL like process to explore solution alternatives[3]. Specifically, this will be useful in the following areas[4]:
    - The role of existing California external filer applications/organizations in the solution, if any.
    - The optimum tools for the business rules engine.
    - The optimum tools for the workflow engine.
    - The optimum tools for the forms/wizard engine.

- The State is not contractually obligated to continue with the current architecture, vendors, or products (e.g., Salesforce).

- In terms of the trade-off between quality, scope, schedule and cost we assume that:
    - The system must have sufficient quality to be functional, reliable, and maintainable at the time it is deployed.

    - A phased deployment approach will be used to deploy incremental capabilities during the CARS restart. This phasing strategy is outlined in the next chapter.

- Based on the results of our 30-day fit-gap analysis, we are assuming that the go-forward strategy does not require that the system integrator use Salesforce as the delivery platform.

- Our Roadmap budgetary estimates, included as part of the WOA #4 deliverable, assume that the Elyon recommendations in this report are implemented.

A more detailed list of assumptions, used as the basis for our developed roadmap forward Rough Order of Magnitude (ROM) budgetary estimates, is defined in Appendix B.

We have not identified any external dependencies related to our recommendations.

[3] We are assuming that a custom cloud-based solution built using a relational DBMS is the correct technical approach. We think that previous SOS experience with attempting COTS transfers has demonstrated that the California requirements in this area are sufficient unique to require a custom solution.

[4] It would also be possible to select the optimal solutions in these areas as part of the acquisition process, potentially with vendor demonstrations as part of the evaluation process.

3 Findings

This Chapter summarizes Elyon's findings, including our observations, recommended phasing strategy, threats and opportunities, and a summary of our overall recommendations. These results are based on our review of roughly 1,500 documents, interviews with almost 50 stakeholders, and application of our eMRI assessment framework. As such, it serves as a summary guiding document to the path forward for the CARS restart. In chapters 4, 5, and 6 we will take a deeper dive into specific elements of the assessment.

3.1 Observations and Conclusions

3.1.1 CARS System Characteristics.

Based on our analysis, we characterize the CARS system as follows:

- Business Functions: CARS business requirements are characterized by complex business rules that are unique to this application and subject to change over time; complex workflow and form-flow requirements that are integrated with the business requirements; significant descriptive, relationship, and financial data that needs to populate to forms and reports; requirements to accept various degrees of "dirty" data for subsequent correction; and extensive version control and redlining at the field level. The application is like a taxation system with requirements for form or wizard-based data entry in accordance with complex requirements, plus compliance reviews/audits. A key requirement of the external community portal is the ability to query, tabulate, and compare data across multiple years.

- Performance: Performance loading is significantly predictable over time, with daily peaks at about 6 PM; bi-annual cycles based on the election cycle; and significant loading near known filing deadlines.

- Security: Data integrity considerations are the primary security concern, with significant consequences in the event of unauthorized data modifications. Versioning to the field level is needed to support internal and external auditing.

- 3rd Party Interfaces: Approximately 70% to 80% of the data input to the system comes from external, third-party vendors. In some cases, this data can be large, consisting of up to a half-million records for a single filing.

- Americans with Disabilities Act (ADA) Capabilities: As with all government systems designed for use by the public, CARS must support ADA accessibility requirements.

- Data Conversion: Historic data, currently in Oracle, must be converted and validated. In some cases, the data will not be compliant with current rules, so for

example current business rules might require an email address, but there will be historic data with no email address and no reasonable way to obtain an email address. So, the data conversion team cannot convert what is not there.

- Maintainability: The system is characterized by business rules, workflows, and form-flows that are subject to change on an on-going basis. It may be necessary to update the central database structures independent of the Application Programming Interface (API) so that the system can be modified while coordinating API changes with the external vendors. The complexity of the business rules means that significant self-test, internal diagnostic, and configurable debug logging capabilities will be needed to maintain the system.

3.1.2 Current project status and conclusions.

The project encountered difficulties with the originally selected implementation vendor (Perspecta), and roughly four years into the effort switched to a new primary implementation vendor, Outreach Solutions as a Service (OSaaS). At the time of the OSaaS contract award, the legislatively mandated deadline for completion of CARS was approximately eight months away. In June of 2021, four-months after the target completion date, the project was paused pending an assessment and development of a go-forward strategy. The CARS project timeline is shown in Figure 2. Perspecta was under contract to deliver CARS from February 1, 2016, through June 30, 2020. Following termination of that contract, OSaaS was under contract to deliver CARS starting on July 1, 2020, through February 28, 2021.

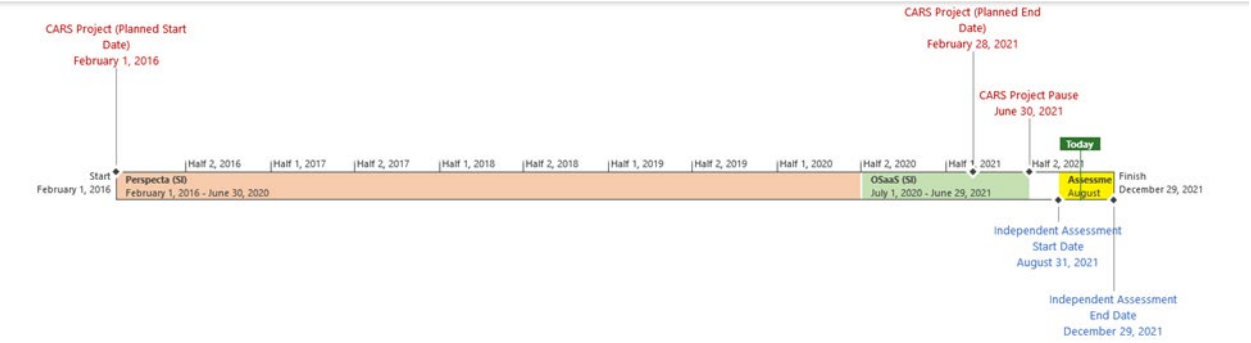

Figure 2: CARS project timeline.

Projects whose project parameters exceed the norm by 50% are characterized by heroic efforts, long hours, burnout, and in the end, failure. The task assigned to the CARS project team overall, of doing several years-worth of development work in eight months, was quite simply impossible.

One characteristic of these projects that are well outside the norm is that time pressure means the vital foundational and architectural work is rushed or skipped altogether. The team moves forward to begin the development phase, without a clear understanding of what needs to be built, and without an optimum and supportable underlying architectural structure. In virtually every case, the project begins to quickly build something, but the

team is building the wrong thing, and they are building it the wrong way. CARS was no exception to the rule.

Information Technology (IT) projects can be thought of as having four layers of functionality (see Figure 3). At the lowest level we have the virtual machine, which is the computer hardware (potentially in the cloud), the database management system, and supporting software architecture. This is the layer where we would decide how the business rule engine would work, how the workflow engine would work, how security access control and monitoring will work, and so on. Most of this layer involves purchasing and configuring items, rather than building them, although some components might need to be built if the organization has unique requirements.

The second level up is the persistence layer, which simply means the place where data is stored. The persistence layer is where the database design comes into play, which then also drives the design of the data objects that will be worked with by the higher layers (e.g., interfaces).

The third level up is the middleware layer. This is the layer where the actual business rules, data validation, workflow configuration, and so on resides. To a large degree, the things that make a business unique are captured in here. If the virtual machine layer is architected correctly, then most of this work involves configuring components with the organization specific data, rather than actual programming. When people talk about business process reengineering, or process optimization, they are mostly talking about changes at this level.

The fourth, and final level, is the presentation layer, or User Interface (UI). This is the computer screens, the reports, the dashboards, and so on.

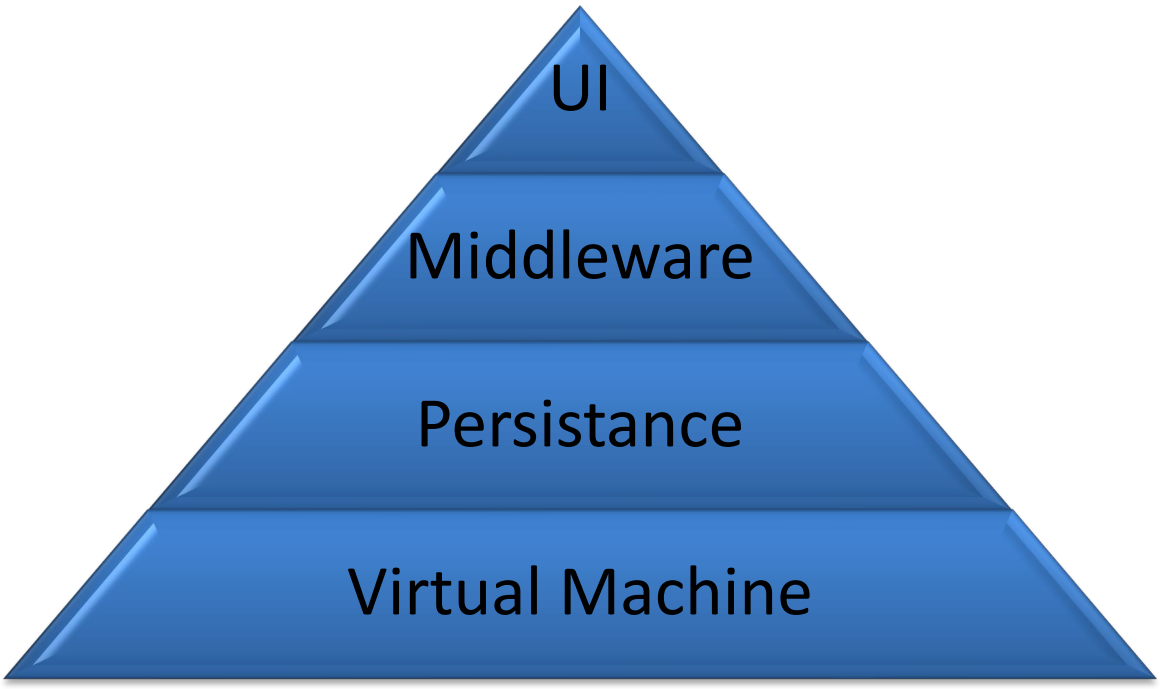

Figure 3: Information technology stack.

Each layer of the technology stack is dependent on all the layers below it. So, problems at the user presentation layer are easy to fix if the layers below are correct. Problems in the middleware layer will require reengineering that layer, but also require changing the presentation layer. This applies all the way down.

In the case of the CARS project, what we've found are serious problems with those lower layers, meaning that fixing the problems cannot be accomplished using a "stay the course" approach, but will require a "salvage and start over" approach.

The current CARS implementation is seriously flawed across all dimensions. There is no rules engine or workflow engine[5]. There is no viable approach to data versioning and error management at the level necessary. All system components are tightly coupled, meaning that changes in one area will have a ripple effect on other areas. The underlying architecture does not meet and cannot meet the system requirements with respect to maintainability and reliability, and correcting these issues will require major work for virtually all system components[6]. The implementation will be difficult to test, is

[5] Instead, the relevant business rules and workflows are hard coded throughout the code base wherever relevant control logic is required.

[6] The existing system consists of a set of Salesforce custom data objects, Salesforce custom Apex code, a Heroku interface that serves the data to a public portal, and Mulesoft API interface. Because the business and workflow logic are embedded throughout the custom Apex code rather than in a data drive business rules and workflow engine, and because the current code only supports a small subset of the required business functionality, that code will need to be completely rewritten. The current data structures are form centric rather than data centric, and again only address a subset of the required functional capability. So those data structures will require significant revision. The changes in the data structures and externalizing the business rules will require that all the forms be revised to integrate with the new

likely to suffer from on-going reliability issues, and will be a major challenge to maintain going forward. The most cost-effective strategy going forward involves reviewing the project artifacts to identify those that are useful, salvaging those components, and then starting over. In many cases, the most useful artifacts will be from the work done during the earlier project evolution (in the 2018 timeframe).

During our initial 30-day assessment, Elyon used quantitative models to assess: the degree to which Salesforce is a match for the given application (the Salesforce fit-gap); the quality of the given Salesforce implementation effort (the Implementation Quality); and the demonstrated capabilities of the given system integration team to perform necessary system integration functions. The result is a score between 1 and 5 where 1 is Very Poor, and 5 is Very Good. The scores produced using Elyon's enterprise maturity readiness index (eMRI) for the CARS project are shown in Table 2 and graphically portrayed in Figure 4. The project has clear challenges in all three areas, and the decision by the Secretary of State to pause the project for an assessment was a wise one. The primary reason that Salesforce is not a good fit for CARS is the complexity of the highly specialized business processes and workflows, and it is the major contributor to the CARS Salesforce fit-gap score of 1.42. In addition, because the PRD business processes, workflows and business rules are specialized and complex, and change fairly often, *and* the current CARS architecture does not address these needs effectively, the resulting implementation quality score is also low.

Table 2: CARS eMRI Scores

	Score (1 to 5)
Salesforce Fit-Gap	1.42
CARS Implementation Quality	2.13
System Integrator Fit-Gap	1.27

architecture. The tight coupling of the portal with the Salesforce data objects using Heroku mean that the portal will need to be completely revamped using the new data structures, plus the portal should be decoupled from the core database anyway for better maintainability. And finally, the new architecture and data structures will require that the API be completely redefined to use the new data structures, and again, to decouple the API from the core database. So ultimately, every aspect of the system will need a major rewrite to fix the current issues.

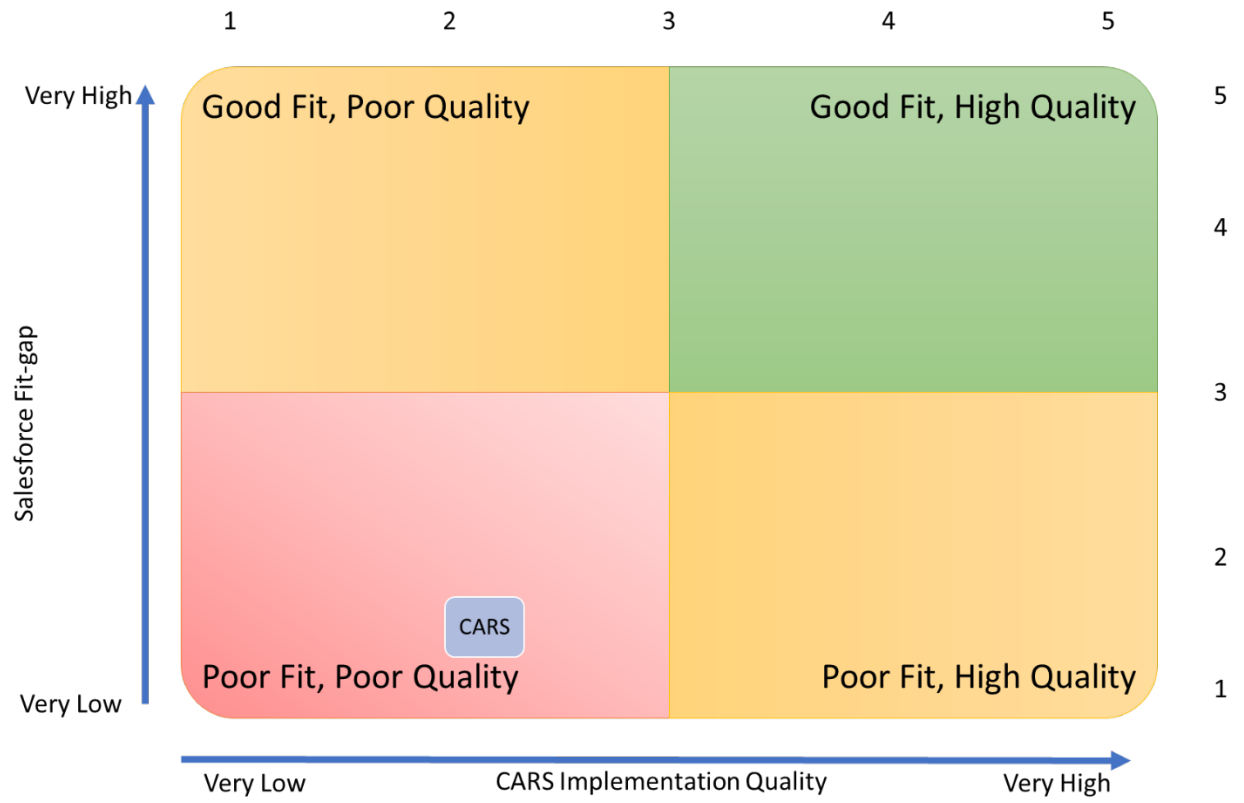

Figure 4: CARS Fit-Gap Quadrant is Poor Fit, Poor Quality

With starting over, there is a serious question as to whether or not Salesforce is the right platform for this application. While it would be possible to use Salesforce for some portion of the required functionality, most of the application functionality will need to be outside of Salesforce. So, it would be possible to begin CARS over again with a different architectural approach, where Salesforce was used for the forms-oriented presentation layer, but external components were then used for the business rules engine, workflow engine, interfaces, and so on. There would then be an internal interface between Salesforce and the remainder of the system. But with this approach there is the follow-on question of whether the functionality that Salesforce would provide is worth the on-going cost of the licenses, the support costs associated with supporting both Salesforce and another solution, and the cost to implement the interface between Salesforce and the remaining system components. Overall, our assessment is that the most cost effective, and lowest risk, approach would be to build CARS without using Salesforce.

Unfortunately, when an IT project has these foundational problems, much of the software that has been developed has limited use. Even when it can be patched and extended to support the updated architecture, it will suffer from problems in areas including reliability, maintainability, security, and performance. For much of the developed application, the total cost of ownership to rework or repurpose the software correctly is typically prohibitive for many reasons (e.g., cost). As a result, it is our recommendation that CARS pivot direction to a solution approach that will result in

lower total cost of ownership of the project life, even though that will require a new acquisition.

The remainder of our report will provide process guidance that will be useful to ensure that the CARS restart is successful.

3.1.3 CARS eMRI Assessment Overview

In conducting our CARS assessment, we used our eMRI project assessment modeling tool. The assessment was conducted for the project as a while, so including both SOS and contractor activities. In other words, we looked at project artifacts that were created without trying to differentiate who created the artifact. The models use a weighted multi-variate assessment approach to arrive at an overall project assessment of 1 to 5, where 1 is Very Low, 3 is typical or average, and 5 is Very High. Table 3 shows the interpretation of the assessment scores in more detail.

Table 3: eMRI Score Interpretation

eMRI Score	Interpretation
5	The project is best in class, with mature and fully supported processes in place covering all major skill areas. Overall, the project performance is best of breed.
4	The project team has effective and fully supported processes in place for most areas, but there are some areas of weakness that may decrease efficiency or increase risk, but not to the point of endangering project success.
3	The project has effective and supported processes in place for key areas, but in other areas the team relies on individual skills and actions. Projects will often get into some trouble, but with work by all members of the team success is achievable. These projects tend to require significant oversight and project/portfolio management attention.
2	The project is largely dependent for success on the skills of individuals doing the work, rather than processes. Small and simple projects will often still be successful, but large and complex projects will have a high failure rate.
1	The project lacks strong processes and is deficient in several important process skills. Project failure is likely, and even smaller projects will often suffer in areas including user satisfaction, maintainability, and cost/schedule control.

Overall, the assessed score for CARS was 2.34. This score was derived from a detailed analysis across the following twelve process categories:

1. CARS Technical Implementation.
2. Data Conversion and Migration, including legacy data migration challenges.
3. Release Management.

4. Testing.
5. Requirement Definition and Management.
6. Project Schedule Management.
7. Risk Management.
8. Contract Management and Vendor Negotiations.
9. Communication Management.
10. Governance and Sponsorship.
11. Organizational Change Management.
12. Quality Management.

The CARS score by process category is shown in Figure 5. Within each process category we evaluate the project in between 3 and 10 Key Process Areas (KPA)s, for a total of 72 separate KPA evaluations. Each KPA is a specific area of work, with an expected outcome (often a document), that we would expect to find. Appendix C documents the detailed eMRI Evaluation Framework that was used, including the 72 KPAs. In the three chapters that follow, we expand on our analysis, looking at factors related to the technical implementation, technical management, and overall project processes.

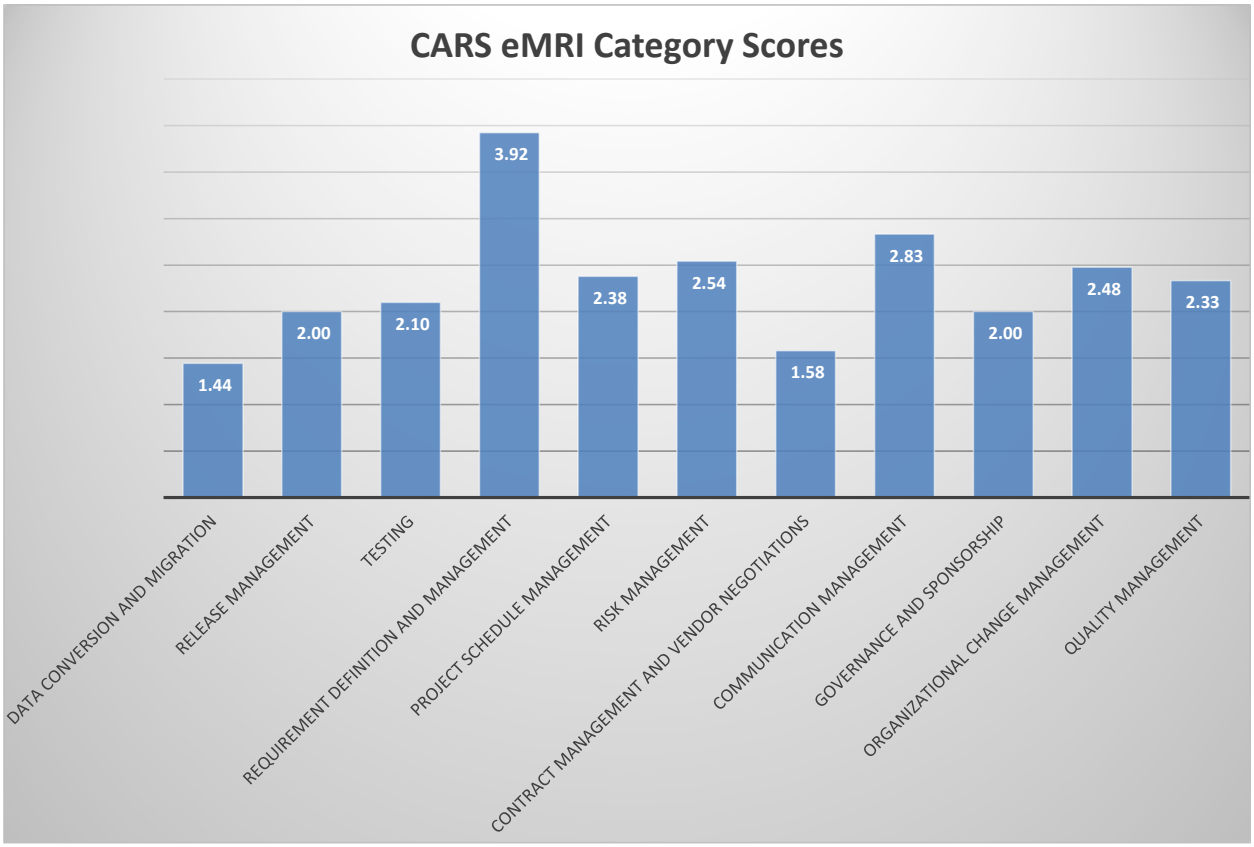

Figure 5: CARS eMRI Category Scores

Our observations and conclusions go across all business functional areas (e.g., Correspondence, Data Retrieval, Filer Disclosure, Filer Registration).

3.2 CARS Restart Phasing Strategy

In developing the specifications for the CARS restart, consideration should be given to a phasing strategy that will deliver some intermediate value-added capabilities as the project is completed. These would be items that offer value to internal or external stakeholders; that could be delivered with no or minimal incremental impact to overall project cost; that are feasible from a business process perspective; and that are technically feasible. For example, based on a workshop conducting with PRD personnel, we believe that the following intermediate deliveries might fit these criteria:

- Early creation of the CARS database. This database could then operate in parallel with the current CAL-ACCESS Oracle Database Management System (DBMS). The database should be structured to allow but tag dirty data. Among other benefits, this would allow the CAL-ACCESS data to be easily migrated to the new database. Following the migration, a refresh capability should be implemented from CAL-ACCESS to the new DBMS.
- Implementation of a public portal to report against the new DBMS.
- Implementation of code to receive electronic files from external vendors using the current file format. A lookup table would then be used to split the data, with some form types loading into CAL-ACCESS and some form types loading into the new DBMS. For example, based on PRD input, the forms shown in Table 4 do not require PRD review, so they might be good candidates to go directly into the new database.
- Installation of the new business rules engine and forms engine. Using those capabilities, new and replacement forms could be created to feed data to the new DBMS.
- Following these early milestone deliveries, the remainder of the CARS capability could be implemented.

Table 4: Forms not requiring PRD review.

Form	
401 Slate Mailer Organization Campaign Statement	511 Paid Spokesperson Report Notification
402 Statement of Termination	615 Lobbyist Report
425 Semi-Annual Statement of No Activity	625 Report of Lobbying Firm
450 Recipient Committee Campaign Disclosure Statement - Short Form	635 Report of Lobbyist Employer or Report of Lobbying Coalition
460 Recipient Committee Campaign Statement	640 Governmental Agencies Reporting
470 Officeholder and Candidate Campaign Statement - Short Form	645 Report of Person Spending $5,000 or More
496 Late Independent Expenditure Report	630 Payments Made to Lobbying Coalitions
497 Late Contribution Report	635-C Payments Received by Lobbying Coalitions
498 Slate Mailer Late Payment Report	E-530 Communications Identifying Candidates Issue Advocacy

3.3 Threats and Opportunities

We have identified the following threats and opportunities for the CARS project going forward:

1. **The sunk-cost fallacy (threat).** It is difficult to mentally set-aside money that has already been spent on CARS as part of the previous two attempts to build the CARS system. There is a natural desire to seek out ways to expand on that earlier work to get value from the funds spent to date. However, forcing the project to use code that is not suitable will result in a CARS restart that is more expensive, less capable, riskier, and that has a higher total cost of ownership.

2. **Vendor underestimation of CARS complexity (threat).** Twice now vendors have looked at the CARS surface (obvious) requirements, failed to appreciate the underlying complexities involved, and bid unrealistic solutions both in terms of the technology/architecture and the proposed level of effort. There is a threat of this happening yet again. This can be partially mitigated with full disclosure of the requirements, but vendors may still look at the requirements quickly as part of a proposal effort and fail to appreciate the underlying challenges. An approach successfully used on both the CDTFA's CROS project and the FTB's EDR and EDR2 projects (both taxation systems) was to incorporate cost realism into the proposal evaluation criteria. Cost realism forces vendors to bid a realistic amount of effort to complete the scope of work.

3. **External filing partners (opportunity):** Most filers in California do their filing through an external filing partner's software. Those vendors have decades of experience with the intricacies of the California filing process, and they have developed software to successfully navigate this landscape. While we do not know the best role for them going forward, there is an opportunity that they may be able to have a role that reduces risk, decreases cost, or decreases schedule.

3.4 Recommendations

In addition to our phased implementation recommendations above, the following sub-sections summarize our recommendations going forward, including the priority of each recommendation. These recommendations are based on benchmark successful projects and are thus agnostic to the specific SDLC or methodology used for implementation. For priorities:

- Critical: These factors were identified as having the highest impact on the success of the CARS restart.

- Highly Important: These factors are important to the success of the project, and failure to implement these may put the project success at risk.

- Important: Implementing these factors will have a significant impact on some combination of risk, cost, or schedule.

Recommendations with respect to schedule, staffing, and budgets will be provided in the WOA #4 deliverable, not in this deliverable.

3.4.1 CARS Technical Assessment.

In this sub-section we will address, critical, highly important, and important recommendations related to:

- Data Conversion and Migration.
- Release Management.
- Testing.

These are areas of the assessment that have a primary impact on the technical staff working on the project. Additional discussion with respect to these areas, and further recommendations, will be found in Chapter Four.

3.4.1.1 Data Conversion and Migration.

Highly Important: CARS should require the implementation vendor to convert the CAL-ACCESS data early in the Software Development Lifecycle (SDLC) for the CARS restart. If the phasing strategy recommended earlier is adopted, the phasing approach will accomplish this automatically.

Highly Important: CARS should require the implementation vendor to develop a data conversion plan that also addresses data validation and ensure that the data validation approach provides the necessary auditable assurance of data accuracy and completeness.

Important: The implementation vendor should be required to create and obtain approval of a security plan that includes data security during the conversion process.

3.4.1.2 Release Management.

Important: The implementation vendor should be required to create and obtain approval of a Release Management Plan for each software release, including phased releases, that describes the release's activities, schedule, resources, roles, responsibilities, risks, and roll-back criteria and strategy.

3.4.1.3 Testing.

Critical: CARS should require a Test Readiness Review (TRR) prior to commencement of System Integration Testing (SIT) and User Acceptance Testing (UAT), and a Production Readiness Review (PRR) prior to release to production. If the phased

deployment approach is implemented, then a TRR and PRR would be conducted for each of the phased releases. These reviews should be gate reviews.

Highly Important: CARS should begin assembling and preparing test cases as soon as possible, with the objective of having most test cases complete by early during unit testing and all test cases complete prior to the start of SIT. Test cases should address normal, exception, and boundary conditions.

3.4.2 CARS Technical Management Assessment.

In this sub-section we will address critical, highly important, and important recommendations related to:

- Requirement Definition and Management.
- Project Schedule Management.
- Risk Management.

These are areas of responsibility that would typically fall primarily on the technical management staff. They cover project planning, execution, monitoring, and controlling. Additional discussion with respect to these areas, and further recommendations, will be found in Chapter Five.

3.4.2.1 Requirement Definition and Management.

Critical: CARS should ensure that non-functional requirements impacting on the technical architecture are called out going forward. Specifically:

- The requirement to accept but tag dirty data for later reporting and clean-up. This may include differentiating between errors and warnings in the data tags.

- The requirement to track version changes to the field level for most data.

- The requirement to easily change business rules and workflows, using data configuration when possible.

Highly Important: CARS should:

- Review all requirement related material, identify the most correct version of the most useful artifacts, and place that material under configuration and version control. A copy should be included in the bidder's library.

- Clarify the role of automation with respect to the requirements (especially the business rules). In many cases, it is currently unclear if a given requirement will be fully automated, partially automated, or remain a manual or external process.

- Clearly differentiate between as-is and to-be processes, where as-is process descriptions may be useful for background information, while the to-be processes are the actual requirements to be verified as part of the new system acceptance.

Important: We recommend that a new requirement management process be implemented as part of the requirement review process recommended above. This requirement management process should include, among other things, requirement related governance.

Important: We believe that there is an opportunity during the requirement refresh to go even farther in terms of potential opportunities for improving the way the PRD business processes work. For example, we believe that by allowing the system to accept dirty data that is automatically tagged with error or warning conditions, automated processes may be put in place to support the cleanup of that data through outreach to the data submitter, through automated processes, or through workflows for PRD analysts.

3.4.2.2 Project Schedule Management.

Critical: Formal estimation techniques should be used when estimating resources and schedule.

Critical: Gate-review milestones should be included in the CARS project schedule. These gate reviews should occur no less often than every six-months during the life of the project. They should include a Software Requirement Review (SRR), Detailed Design Review (DDR), TRR, PRR, and a Post Implementation Evaluation Review (PIER). These reviews are covered in more detail in Chapter Five.

Important: Resource leveling should be used to ensure that project resources are not overloaded.

3.4.2.3 Risk Management.

Important: Risk managers should actively seek and accept risk related input from all stakeholders, both internal and external. Risk input should be utilized no matter what approach is used to communicate the risks (e.g., email, spreadsheets, phone, verbally).

3.4.3 CARS Process Assessment.

In this sub-section we will address, critical, highly important, and important recommendations related to:

- Contract Management and Vendor Negotiations.
- Communication Management.
- Governance and Sponsorship.
- Organizational Change Management.
- Quality Management.

These areas are often the responsibility of the government, often with the assistance of vendors that are independent of the system integrator team. Additional discussion with respect to these areas, and further recommendations, will be found in Chapter Six.

3.4.3.1 Contract Management and Vendor Negotiations.

Critical: The CARS recompete acquisition should be structured as a competitive firm fixed price deliverable-based contract, thereby allocating the risk to the party best able to manage that risk (the vendor). Work where the level of effort is largely under the control of the SOS, specifically in the areas of training, transition, maintenance, and operation may best be structured using fixed labor hour bids.

3.4.3.2 Communication Management.

Highly Important: A project health dashboard and vendor performance balanced scorecard should be developed and updated on a regular basis.

Important: A project website should be established and maintained to communicate project status and information to internal and external stakeholders.

3.4.3.3 Governance and Sponsorship.

Critical: A Project Charter should be created and approved. A Project Sponsor should be identified and be given the responsibility and authority necessary for project success, including the final decision authority with respect to moving forward beyond each gate review.

3.4.3.4 Organizational Change Management.

Critical: After selection of the technology solution as a result of the acquisition, a skill gap analysis should be conducted for all stakeholders, internal and external, and including business, executive, technical, and support stakeholders. This gap analysis should be used as a primary input to the OCM and training plans.

3.4.3.5 Quality Management

Highly Important: All project artifacts, including hardware, software, code, engineering artifacts (documents), process documentation, and requirements should be placed under configuration control, with one approved and most current version.

Highly Important: An IV&V vendor should be brought on-board to focus on verification and validation of contractual products, including internal and external consistency, correctness, and fit-for-purpose.

Highly Important: Project processes should be documented, optimized, and approved.

Important: A quality assurance group should ensure that project processes are correctly followed.

4 CARS Technical Assessment

In this chapter we cover the CARS technical assessment. This area encompasses three major capabilities in the eMRI model:

1. Data Conversion and Migration.
2. Release Management.
3. Testing.

Within each dimension, CARS was assessed with a score of 1 to 5, where 1 would be highly ineffective, 3 would be moderately effective (average or typical), and 5 would be highly effective. As shown in Figure 6, CARS was somewhat above average in requirement definition and management area, but significantly below average in project schedule management and risk management. In the remainder of this chapter, we will provide additional details in each of these areas.

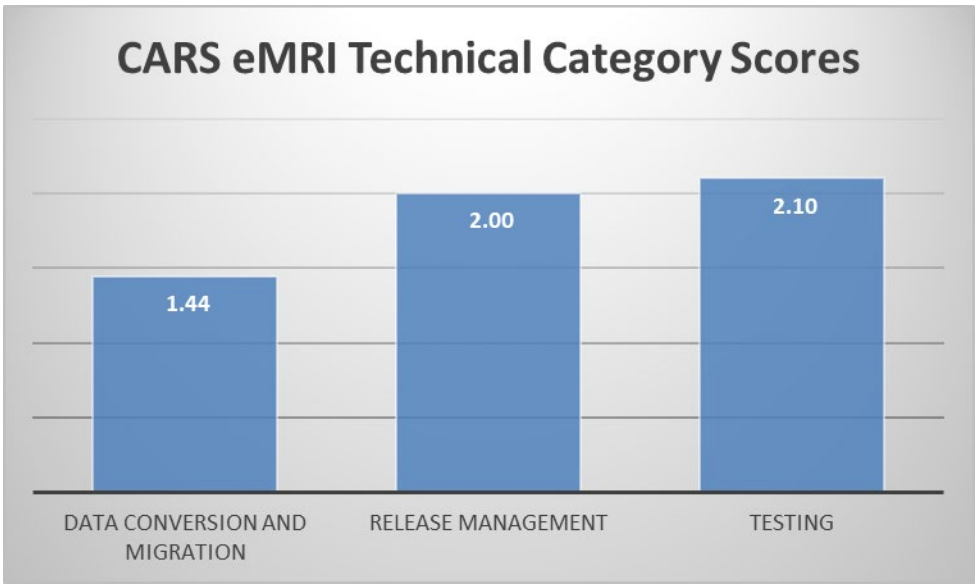

Figure 6: CARS eMRI Technical Category Scores

4.1 CARS technical implementation.

As discussed in earlier chapters, the existing CARS implementation is flawed at the architecture, data structure, middle-tier, and user presentation layers. The architecture, data structures, and middle tier are not correctly optimized for the necessary business processes, workflows, and data structures. For example, there is no business rules engine or workflow engine. The external components (portal and interfaces) are tightly coupled to the core data structures. The data structures cannot accept dirty data and have limited versioning capability. The user presentation layer takes a purely form-centric view of the world, as opposed to an underlying data-centric view. Those flaws are fundamental and recasting the project will be more cost effective than to continue

development. Anything less will likely result in an unreliable system with significant functional deficiencies that is expensive and difficult to maintain.

For IT systems in general, preparing and approving requirement and high-level design documentation represents a significant portion of the effort. Much of the existing CARS work in this area should have a high degree of reuse. Similarly, project management and organizational change management documents should largely carry forward to the CARS restart, with simple updates. There have been two mostly successful CARS data conversion efforts (one with Perspecta and one with OSaaS), and the business rules used for data clean-up and transformation during those efforts should be largely reusable going forward.

4.2 Data Conversion and Migration.

In this Section we cover both legacy data and data conversion. Data migration is a critical component of all IT modernization projects. While it is often thought of as an activity that occurs at the end of the project as part of cut-over, that final conversion is only the tip of the iceberg. Data conversion is always high risk with a lot of uncertainty about data quality, so it should be an early focus of the project team. In addition, an initial version of the full production database should be available to support unit testing (for boundary testing and error handling, in particular) and to support performance testing. Fortunately, most of the data in any legacy system is static. Static data can be converted at any point in the SDLC, greatly simplifying the final cut-over data conversion.

For systems that demand a high degree of data validation, such as a banking system, data conversion will require a significant amount of effort to find and fix any errors or inconsistency in the data. The design concept here is to enforce data integrity at the point of entry, so that subsequent system components can rely on the quality of the data in the database. However, there is an alternate design strategy that allows dirty data into the database, but then tags that data with an error or warning and severity. Subsequent system components must then be designed to watch for those flags and be aware of the potential data problems. This approach is preferred in situations where data quality will be incrementally improved over time through automated or manual review processes. For example, suppose someone is completed a form (a data record) and one of the required fields is the organization’s Tax ID (TID). One strategy is to not accept the record submission without that required field. So, if a person fills out the form almost completely, gets to the TID field, then realizes that they do not have the TID and must obtain that information elsewhere, the partial record is not accepted into the database. Under the alternate strategy, the available data would be accepted and the TID field would be flagged as an error (missing data). Workflows could then be put in place to correct the record. So, for example, an email might be sent asking the filer to update the form with the Tax ID.

One advantage of accepting but tagging dirty data is that it greatly simplifies the data conversion process. Dirty data can be converted but tagged, and automated or manual processes can then work to clean up the data after conversion. A disadvantage to the

dirty data approach is that system processes must be aware that the data may be dirty, so there is added coding logic required to handle those exception conditions.

According to Interviews and data conversion related status documentation, data is currently being converted directly from the legacy system into the CARS Salesforce data objects, with clean-up occurring in the form of transformations during the Extract-Transform-Load (ETL) process. Because the current data structures will need to be re-architected, there is very little value in this Salesforce stored data. However, business logic and code needed to clean the legacy data will be useful to the project going forward under any approach, because that legacy data will need to be cleaned under all scenarios. There is potential value of the current (OSaaS) data conversion work based on the business logic used for the transport and data cleanup operations.

Additionally, we did find that there was significant data conversion and migration work performed by the CARS project back in roughly 2018, and in reviewing that work, it appears that there may be useful design work, and potentially useful converted data, from that timeframe[7].

CARS overall score in the data conversion area was 1.44. Figure 7 shows the score for each of the eMRI data conversion Key Process Areas (KPA)s. We will address each KPA individually in the sections that follow.

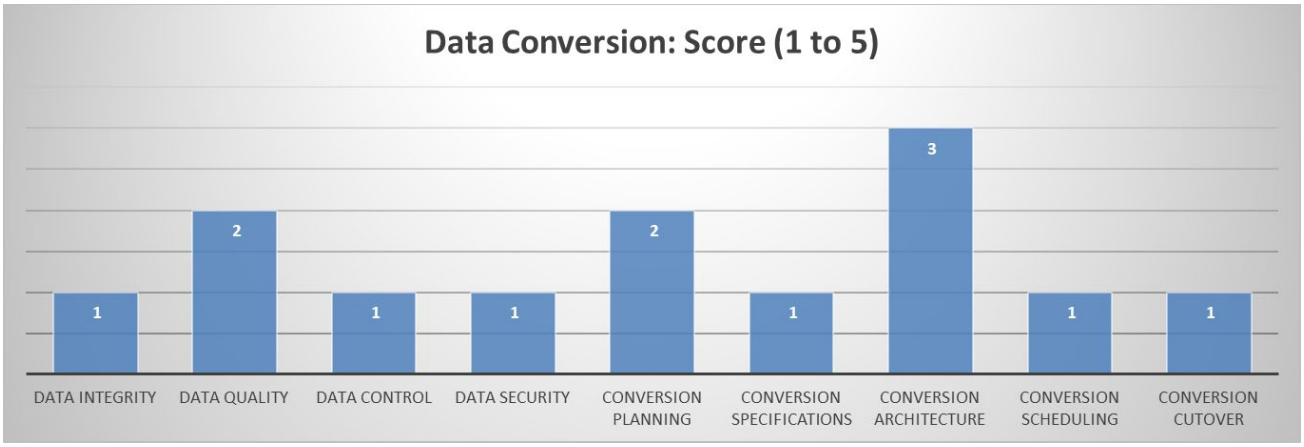

Figure 7: Data conversion KPA scores.

4.2.1 Data Integrity.

Data integrity during data migration involves ensuring that the migrated data is provably reflective of the source (legacy) data, with any changes to the data fully documented and approved by the data owner. This requires verification that 100% of the source data was migrated, with no data loss or unplanned data changes. This migration should be auditable, in that a knowledgeable third party can review the conversion documentation and be satisfied that data integrity requirements were met. Note that the simple lack of errors during conversion does not meet this requirement. Secondary checks must be

[7] Specifically, the artifacts found on the SOS SharePoint site, previous project directory, within the folders Data Cleanup and Data Conversion.

used to validate that no data was lost or unintentionally modified during the conversion process. If sampling will be used (as opposed to full verification), then a risk assessment should be performed to evaluate the organization vulnerability in the event of intentional or unintentional data changes, and a suitable sampling strategy applied to achieve a risk appropriate degree of confidence.

In the case of the CARS project, we did not find evidence of any attempt to verify data integrity as part of the data migration, so we assigned a score of Very Low. We do not believe that there are any reusable artifacts in this area applicable to the CARS restart.

4.2.2 Data Quality.

In Wikipedia, data quality is defined as follows:

> . . . data is generally considered high quality if it is "fit for [its] intended uses in operations, decision making and planning".[8], [9], [10][1][2][3] Moreover, data is deemed of high quality if it correctly represents the real-world construct to which it refers. Furthermore, apart from these definitions, as the number of data sources increases, the question of internal data consistency becomes significant, regardless of fitness for use for any particular external purpose.[11]

After successful data conversion, we would like to see all data migrated, cleansed and usable with duplicate data resolved and removed. In the case of systems that allow dirty data (as we recommend for CARS), it is acceptable for data to be migrated with errors if those errors are correctly flagged in the database. This approach allows the data to be cleaned gradually over time, and some historic data might be left in a known error status indefinitely.

In the case of the CARS project, both data conversion efforts (Perspecta and OSaaS) attempted to clean the data as part of the migration, and both were apparently largely successful (based on the data conversion status reports). However, we did not find any post conversion analysis of data quality, and in reviewing the data conversion rules it appears that in many cases “dummy” data was inserted into records, or records were

---

[8] Redman, Thomas C. (30 December 2013). Data Driven: Profiting from Your Most Important Business Asset. Harvard Business Press. ISBN 978-1-4221-6364-1.

[9] Fadahunsi, Kayode Philip; Akinlua, James Tosin; O’Connor, Siobhan; Wark, Petra A; Gallagher, Joseph; Carroll, Christopher; Majeed, Azeem; O’Donoghue, John (March 2019). "Protocol for a systematic review and qualitative synthesis of information quality frameworks in eHealth". BMJ Open. 9 (3): e024722. doi:10.1136/bmjopen-2018-024722. ISSN 2044-6055. PMC 6429947. PMID 30842114.

[10] Fadahunsi, Kayode Philip; O'Connor, Siobhan; Akinlua, James Tosin; Wark, Petra A.; Gallagher, Joseph; Carroll, Christopher; Car, Josip; Majeed, Azeem; O'Donoghue, John (2021-05-17). "Information Quality Frameworks for Digital Health Technologies: Systematic Review". Journal of Medical Internet Research. 23 (5): e23479. doi:10.2196/23479. PMC 8167621. PMID 33835034.

[11] https://en.wikipedia.org/wiki/Data_quality.

changed using unapproved rules, to force the data into the target database. Overall, we assessed this KPA at Low.

The OSaaS data conversion effort moved data directly into Salesforce. Because the CARS Restart will not require Salesforce as the platform, the data conversion transformation rules may be reusable going forward but the data itself is not useful. However, there was a significant data clean-up effort undertaken by Perspecta in 2018, and that data was stored in a standard relational database. That cleaned data may be usable for the CARS restart. In either case, if the data transformation rules were not approved by the data owner(s), then those rules should be reviewed and approved as part of the restart (see the discussion below regarding Data Control).

4.2.3 Data Control.

Data control is the process of governing and managing data[12]. Audit requirements dictate that the data owner (also called the data steward) must always maintain control of the data. That is the reason that manual changes to data should always be made by the business, not by the information technology staff. And while IT staff do change the data indirectly through the application of automated rules during data transformation, those rules should be reviewed with and approved by the data owner. So, all controls and business rules that automatically change data must be validated by the data owner(s), validation should be documented, and performance of those transformation rules should be measured for correctness.

In the case of the CARS project, we assessed performance in this dimension as Very Low, in that data transformation rules were created ad hoc by the developers as part of the conversion work without adequate review and approval by the business data owner. We do not believe that there any reusable artifacts in this area going forward.

4.2.4 Data Security.

Data security during migration is a significant area of vulnerability. While production data is often protected through access controls, encryption of data at rest and in motion, audit logs, and automated intrusion detection systems, that same data during conversion may be completely exposed and vulnerable, with an organization not even having the ability to detect unauthorized changes. For that reason, a project's security plan should specifically address data security during migration, and suitable controls should be put in place to ensure the integrity of, and control access to, that data. In general, data at rest and in motion should be under controls equivalent to those used in the production system, including logging and monitoring.

[12] For a good discussion of this topic, see https://simplicable.com/new/data-control#:~:text=Data%20control%20is%20the%20process%20of%20governing%20and,the%20accuracy%2C%20completeness%2C%20credibility%20and%20timeliness%20of%20data.

In the case of the CARS project, we did not see data conversion specific security safeguards implemented, and we assessed the project as Very Low in this KPA. We do not believe that there are any reusable artifacts in this dimension.

4.2.5 Conversion Planning.

An approved data conversion plan should exist covering conversion strategy, assumptions, constraints, activities, resources, roles, responsibilities, timeline, milestones, and risks.

In the case of the CARS project, data conversion documentation (including a Data Conversion Plan) was prepared during the Perspecta work on the contract but was apparently not updated for the OSaaS effort. Based on the existence of those earlier documents, we assigned CARS a score of Low in this KPA. Those earlier Perspecta documents might be reusable as a starting point for the conversion planning efforts going forward.

4.2.6 Conversion Specifications.

An approved data table conversion specification should exist documenting target and source data tables with mapping, data volume (e.g., record counts), field mappings, transform rules, validation requirements, and differentiating static from dynamic source data. Examples of transformations include cleaning, filtering, validation, splitting, joining, derivations, and applying specific business rules. Identifying specific rules for tagging dirty data with an error or warning would be included as part of conversion specifications. This specification should also explicitly document the legacy data tables that will not be converted, typically including archival, obsolete, and log tables. While we did not find an OSaaS document covering this area, there is a Perspecta document, the “CARS Data Standardization Cleansing Plan_100518_.docx”, that may be useful as a starting point. We assigned a score of Very Low in this KPA.

4.2.7 Conversion Architecture

All large modernization projects should include a separate data conversion environment as part of the overall system architecture. This intermediate staging environment allows the data transformations to occur outside of the other environments (e.g., dev, test, production). We also recommend that data be transformed in this separate environment, and then loaded into the production database.

In the case of the CARS project data was migrated directly into Salesforce, which we do not recommend for large and complex systems. Overall, CARS was assigned a score of Average in this area, in that the project used standardized conversion tools and resources to perform their data conversion work. We do not believe that there are any reusable artifacts in this KPA.

4.2.8 Conversion Scheduling

Because data conversion is a risky and difficult endeavor, it should begin early in the SDLC. An initial data conversion pass should ideally be completed prior to the start of unit testing, but certainly prior to performance testing during SIT. Most converted data will be static, so that static data can be converted at any time. Dynamic data will need to be refreshed or updated at the time of final cutover to capture recent data changes.

In the case of the CARS project, conversion was continuing during UAT, which because a major barrier to effective testing. We assigned a score of Very Low in this KPA. We do not believe that there are any reusable artifacts in this area.

4.2.9 Conversion Cutover

Cutover is the process of performing the final database conversion/synchronization, verifying that the new system is functioning as intended, and shutting down the legacy system. During the cutover window, the legacy system will be unavailable. In most cases, neither the legacy system nor the new system are available during the cutover window, and users will simply see an “Unavailable because of scheduled maintenance” screen. As part of the conversion plan, the business owner should define the duration of the cutover window, which can be as short as seconds or as long as weeks. A weekend, or extended holiday weekend, is often designated as the cutover window. Because the legacy system is shutdown during that cutover window, final data conversion is possible. Final cutover conversion will ideally require 25% or less of the available cutover conversion window, thereby allowing some recovery time, plus time to verify proper system operation after cutover. In addition, cutover success and failure criteria should be defined, and an approved rollback strategy should be in place.

In the case of the CARS project, we did not find any document that specifically addressed the conversion cutover strategy, so a score of Very Low was assigned. We do not believe that there are any reusable artifacts in this area.

4.3 Release Management.

Release management is an important component of a successful system deployment and subsequent transition to M&O, but for phased or Agile development it becomes important earlier in the SDLC as various parts of the final solution are released to production. At a high level, release management involves planning for the release, documentation of the release (e.g., new features), and the integration and testing of the new or modified system components. Overall, our eMRI assessment of CARS release management capabilities was 2.0 out of 5. The specific scores for each of the three-release management KPAs is shown in Figure 8. Each of these areas will be discussed further in the sections that follow.

Figure 8: CARS release management KPA scores.

4.3.1 Release Planning.

Each major software release should include an approved Release Management Plan that describes the release's activities, schedule, resources, roles, responsibilities, risks, and roll-back criteria and strategy. Activities defined in the plan should normally include:

- Management.
- Release Planning.
- Test Planning.
- Integration test.
- System test.
- Regression testing.
- Documentation.
- Release and deploy.

In addition, ADA compliance testing should be included in the release strategy. Although full ADA testing is not typically performed with every release, some level of ADA testing for new components should be included in the strategy, with full ADA compliance testing on a regular basis (e.g., annually).

As a best practice, the group responsible for release management and the group responsible for development are frequently separate. This provides a degree of independence both for release testing and for the decision whether to release software to production.

We did not find evidence of CARS creating a formal Release Management Plan, so the project was assigned a score of Very Low. It appears that release planning was more ad hoc, described as part of standard project plans and status reporting. We

recommend that a Release Management Plan be included as a formal deliverable for the CARS restart and suggest establishing a group separate from the developers with responsibility for release related activities.

We did not find any release management planning materials that we believe are useful for the CARS restart.

4.3.2 Release Documentation.

Each release should include documentation tailored for that release, including a Release Management Plan; a Release Test Plan; and a Release Test Report. A version description document should be prepared for each release, describing that version of the software, and including changes from previous versions. Technical and user documentation should be updated with each release to remain current.

In the case of the CARS project, there were no formal software releases to production, so this KPA could not be fully assessed. However, it appears that the release related documentation that was under development was informal and distributed amongst other documents. There were test reports, but it was not clear what components of the testing pertained specifically to the release readiness. We assigned a score of Low in this category. Going forward, we recommend that the CARS restart require release specific documentation, as described above, for each major software release.

We did not identify any release documentation that would be useful for the CARS restart.

4.3.3 Release Regression Testing.

Each release should include full regression testing, plus ADA compliance verification for new components. Regression testing should be fully or mostly automated using regression test scripts, and ADA compliance verification should use tool-based compliance validation.

CARS did conduct regression testing, using partially automated regression test scripts. ADA compliance was tested, although the project did not make any attempt to remediate the ADA problems that were discovered by the tool. We assigned a score of Average in this category. We did not identify any release regression testing materials that would be reusable to the CARS restart.

4.4 Testing

Roughly one-third of total project effort is typically spent on test related activities. Based on the interviews and the test related documentation reviewed, we found that CARS testing was flawed in almost every way that it could be flawed[13]. There were no

[13] See for reference to testing best practices, ISO/IEC/IEEE 29119, ISO/IEC 9126, IEEE 829, and IEEE 12207.

consistently agreed to goals and objectives that the system could be tested against. There was no realistic test data set or correct test scripts. Unit testing was inadequate. System Integration Testing to verify proper system operation prior to User Acceptance Testing was either skipped or so inadequate that the effect was the same. There was little or no regression testing. ADA testing was an afterthought, and there was no attempt to resolve ADA issues. No Test Readiness Review milestone was conducted. User acceptance testing (internal and external) did not have sufficient time, clearly defined roles, or objectives. For example, we were told that external testing was scheduled for two-weeks, and that the first scheduled week was during the busiest filing week of the year, so no external testers were available. Then during the remaining week, external testers told us that it took them three days to receive the credentials needed to be able to login, and that during the remaining two-day testing window the system kept locking up, requiring them to call the developers to have them manually clear errors. Ultimately, the people best able to test the system simply gave up.

Our eMRI overall score for CARS testing was 2.1, with the detailed KPA scores shown in Figure 9. Each of these areas will be addressed in further detail below.

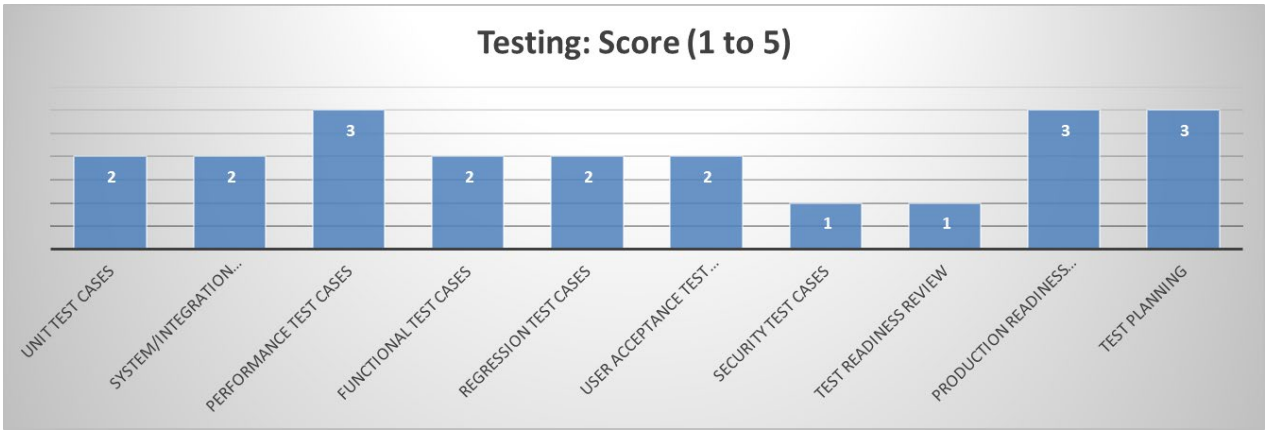

Figure 9: CARS testing KPA scores.

4.4.1 Unit Test Cases.

Effective unit testing lays a foundation for success at all other stages of testing. However, developers are often expected to create their own unit test cases, and this is seldom effective. Instead, the test cases that will be used during later stages of testing should be created by the Business Analyst and SMEs early enough in the SDLC to be used by the developers for unit testing. This will allow the developers to verify that the software does what will be expected prior to involving the Subject Matter Experts (SME)s. Done right, the only defects that should be discovered during SIT and UAT should be true integration errors, and cases of the developer misunderstanding the intended outcome for a given test case. So, the best practice is for a comprehensive and approved set of unit test cases to exist and be available to the developers prior to developing each functional area.

One common deficiency is to focus on the procedural aspects of test case creation, and to ignore or minimize the data aspects of test case creation. In fact, appropriate and standardized test data is every bit as important as the test steps and intended outcomes. So, a standardized set of unit test data should be developed to accompany the test cases, and again, this should be available prior to the start of unit testing.

Another common deficiency is to focus the test cases exclusively on the intended or expected logic paths through the system. In fact, between 50% and 75% of the test cases should expose boundary (e.g., a maximum length string) and exception (e.g., an invalid date) conditions.

CARS scored Low in this category, indicating that the developers were responsible for creating their own unit test cases. We believe that some of the unit test scripts may be reusable going forward. The CARS unit test scripts should be evaluated to determine which of them are usable for the CARS restart.

4.4.2 System/Integration Test Cases.

System Integration Testing verifies that system components are interoperating as intended. A comprehensive and approved set of SIT test cases, along with appropriate test data, should be available prior to the start of SIT testing. SIT testing should include interface testing using realistic load and boundary testing for interface capabilities.

In the case of the CARS project, a score of Low was assigned. Some of the CARS test cases would be classified as SIT test cases, and there were some interface related test cases. However, the SIT test cases were not comprehensive, there was little or no accompanying test data, and interface testing did not include realistic load and boundary testing.

We do not believe that any of the CARS SIT test cases will be reusable going forward, because of likely architectural changes.

4.4.3 Performance Test Cases.

Performance testing is an often-neglected area of testing that may then result in a production system with unacceptable performance characteristics. The key to effective performance testing is two-fold. First, the processes that are likely to be the performance bottlenecks need to be identified. Second, effective performance testing requires performance test datasets that are at least as large as the actual data that will need to be searched or processed. So effective performance testing involves performance analysis, development of a comprehensive and approved set of performance test cases, and creation of a performance test dataset that is similar in size and characteristics to the production data. This data may be generated, or it may be an actual copy of the production data. If actual production data is used, then it is best practice to sanitize sensitive data. If performance testing is used to size servers, then a 50% Central Processing Unit (CPU) reserve capacity should be planned for (this does

not apply for cloud-based deployments, which can have increased capacity as required.)

In the case of the CARS project, we saw some limited examples of performance testing in the area of large file imports, but no comprehensive and approved set of performance test scripts and data. We assigned a score of Average in this KPA.

We believe that the data conversion work discussed above will be useful going forward in terms of creating the basis of a set of performance test data.

4.4.4 Functional Test Cases.

Functional testing is where the business functionality is verified. A comprehensive and approved set of functional test cases should be available prior to the start of UAT. Equally important, functional test data needed to fully test the system functionality, and with known characteristics, should be available to accompany the test cases. Functional testing should include functional, boundary and exception testing.

In the case of the CARS project, a score of Low was assigned. Functional test cases were available, however there was limited or no suitable test data available. In addition, test cases focused primarily on the expected path rather than defining the boundary and exception test conditions that should be verified.

We believe that some of the functional test cases may be reusable going forward. The CARS unit test cases should be evaluated to determine which of them are usable for the CARS restart.

4.4.5 Regression Test Cases.

Regression testing is important because changes or fixes to one part of the application might break a part of the application that has already been tested. Regression testing ensures that the things that were working before are still working. Because regression tests are performed over and over both during acceptance testing and following release to production, the additional effort to automate these tests is generally justified. Regression test automation normally begins during SIT/UAT, and those automated scripts and data are then used as the starting point for the automated scripts that are developed and maintained by the release team.

CARS did conduct regression testing, using partially automated regression test scripts. We assigned a score of Low in this category. We did not identify any regression testing materials that would be reusable to the CARS restart.

4.4.6 User Acceptance Test Cases.

While functional testing verifies system functionality with a fine level of granularity (e.g., testing the functionality of the individual fields on a screen), UAT test cases will verify

the system's ability to do meaningful work from a business user perspective. So, these test cases would typically be at the level of workflows, use-case scenarios, business processes, and so on. A full set of approved UAT test cases with corresponding data should be available prior to the start of UAT.

In the case of the CARS project, we assigned a score of Low. Some UAT test cases were available, but they were not comprehensive or fully approved, and they did not include the corresponding data.

We believe that some of the user acceptance test cases may be reusable going forward. The CARS user acceptance test cases should be evaluated to determine which of them are usable for the CARS restart.

4.4.7 Security Test Cases.

Effective security testing begins with a vulnerability assessment during which the system, and in particular the data, is reviewed to identify the likely threats and the severity of different types of security incident. Based on this review, a security risk mitigation strategy is put in place and security test cases are developed to verify that strategy. The result will be a comprehensive and approved set of security test cases that are available prior to the start of SIT. Test cases should cover all aspects of system security, both physical and logical. Test cases should include coverage for vulnerabilities including confidentiality, integrity, authentication, authorization, availability, and non-repudiation. A National Institute of Standards and Technology (NIST) compliant standard testing framework such as the Safeguard Computer Security Evaluation Matrix (SCSEM) should be used to ensure comprehensive coverage of security related issues.

In the case of the CARS project, the assigned score was Very Low. We did not find evidence of any organized or comprehensive approach to security analysis and testing, and our analysis of the current CARS code did point to the presence of security flaws.

We do not believe that there are any reusable components in the security testing area for the CARS restart.

4.4.8 Test Readiness Review.

SIT and UAT are disruptive to an organization, because they involve support by a significant number of SMEs from the business unit(s) involved. It's important that the testing process only begin when the system is fully ready for testing and certification. Starting prematurely is expensive in terms of labor, is frustrating for users, and it prevents the discovery of subtle defects because the SMEs are dealing with the more glaring problems. To avoid these problems, best practice is to hold a formal gate review, normally termed a Test Readiness Review, and to only commence SIT and UAT after successfully passing this milestone.

In the case of the CARS project, there was no TRR, UAT was commenced prematurely, and the results were as described above. We assigned a score of Very Low to this item. We do not believe that there are any reusable components associated with this KPA going forward.

4.4.9 Production Readiness Review.

Something worse than entering SIT and UAT prematurely is deploying a system prematurely. In the case of a premature deployment to production, the impacted stakeholders include not just the business unit, but also external stakeholders. Severe post deployment problems can result in long-term damage to an organization's reputation and resolving data issues that may result from those deployment issues can be difficult and costly. For these reasons, best practice is to hold a formal gate review, normally termed a Production Readiness Review, and to only deploy to production after successfully passing this milestone.

In the case of the CARS project, the project did not get to the point where a PRR would have been held, however based on the planning documentation we assessed this area as Average. We do not believe that there are any reusable components associated with this KPA going forward.

4.4.10 Test Planning.

The controlling document for testing is the Software Test Plan (STP). The STP is a comprehensive and approved plan that covers testing activities, schedule, roles, responsibilities, criteria, and resource requirements. It should fully document all phases of testing from unit testing through release to production.

In the case of the CARS project, an STP was prepared, and testing related resources were estimated. However, the STP was not comprehensive, and the resource estimates were prepared relatively late in the process, thereby creating scheduling problems. We assigned a score of Average in this KPA. We believe that the STP could be updated and reused going forward into the CARS restart, although the resource estimates will need to be significantly updated.

5 CARS Technical Management Assessment

In this chapter we cover the CARS technical management assessment, which encompasses the major areas of work that are the focus of the project's technical management team. This area encompasses three major capabilities in the eMRI model:

1. Requirement Definition and Management.
2. Project Schedule Management.
3. Risk Management.

Within each dimension, CARS was assessed with a score of 1 to 5, where 1 would be highly ineffective, 3 would be moderately effective (average or typical), and 5 would be highly effective. As shown in Figure 10, CARS was somewhat above average in requirement definition and management area, but below average in project schedule management and risk management. In the remainder of this chapter, we will provide additional details in each of these areas.

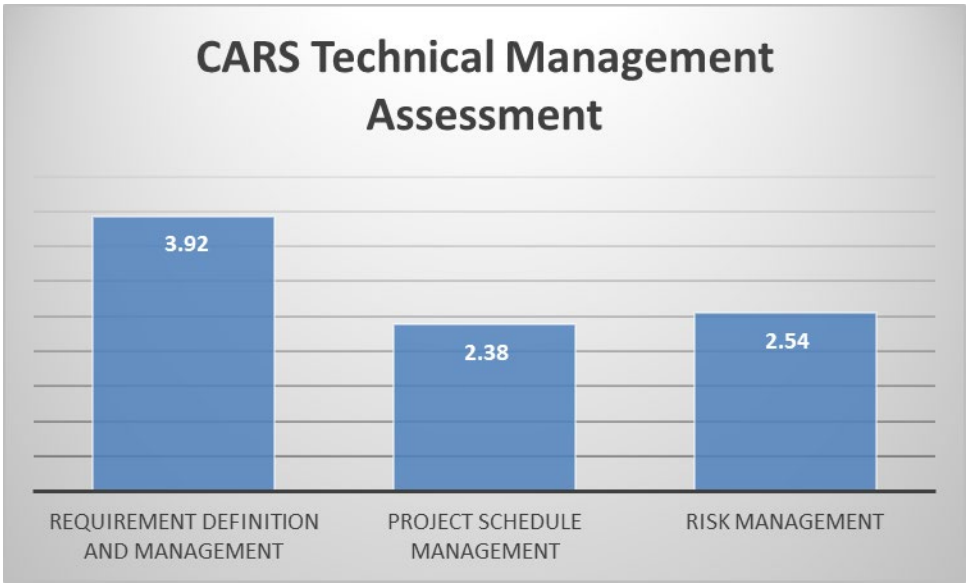

Figure 10: CARS Technical Management Assessment Results

5.1 Requirement definition and management.

An effective requirement definition and management process is arguably the most critical component of a successful IT project. Requirements form the basis of effective estimation, scheduling, detailed design, testing, governance, and many other elements of the implementation project. As shown in Figure 11, the CARS requirement definition and management process was effective, and in fact, was above the average score of 3 in most dimensions. The CARS project challenges were not caused by a lack of good requirements.

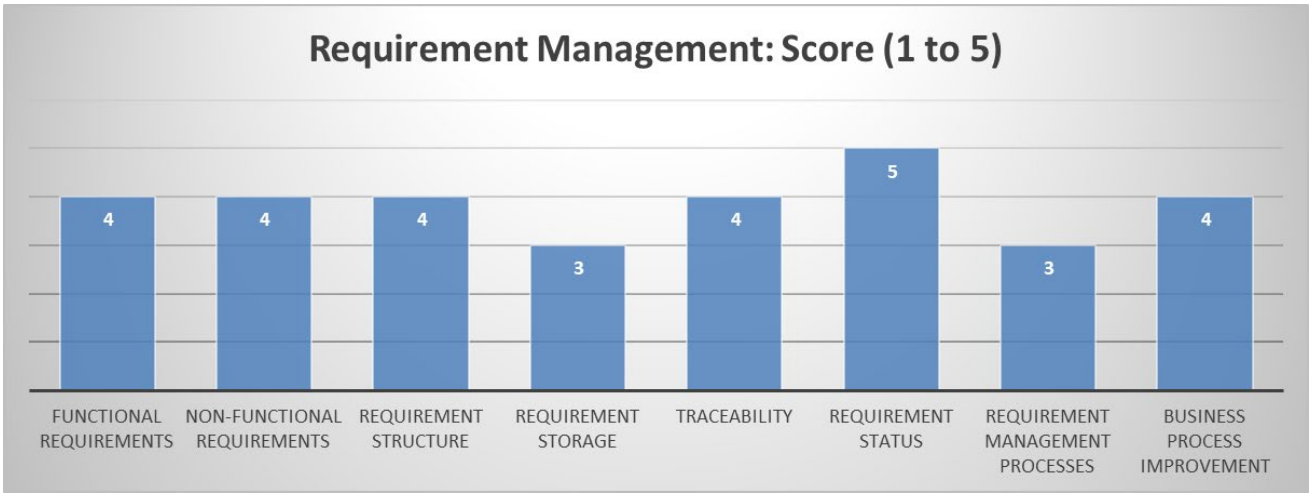

Figure 11: CARS Requirement Definition and Management Assessment

For complex IT systems we recommend a three-tier approach to thinking about and developing the system requirements, as follows.

1. **Tier-one:** This tier represents a strategic perspective of the requirements. Why are we doing this? What do we need to fix? What are the underlying sources of requirements (e.g., legislation)? What's the mission of the organization, and how does this system support that mission? How are these areas clustered (e.g., Campaign versus Lobby)? These strategic requirements set the overall direction for the entire project.

2. **Tier-two:** This tier captures the business requirements. These are the organization's actual business needs. They are captured in the form of high-level business objects, each of which represents a discrete deliverable component that can be defined, built, tested during UAT, and demonstrated. Examples include workflows, use-case scenarios, interfaces, reports, and screens (a more complete list is shown in Table 6 below). These define the business functions that will be accomplished using the system. They should be clustered based on business process area, so that the implementation team can work with one group of SMEs at a time to understand their work. Business rules fit into this category because they are business centric requirements. The business requirements should be fully defined prior to conducting an acquisition, and they should be incorporated into the resultant contract.

3. **Tier-three:** This tier captures the technical requirements. These are the specifics of how each interface will work, what will be on each page and report, and so on. These will be what the programmers need to code, and what the testers need to conduct unit testing. These are derived requirements, in that they are derived from the higher level, business requirements. These should be defined jointly by the business and the implementation vendor because they will be based on a combination of business requirements and the implementation specific approach. In most cases, it's a mistake to try to define these in final form prior to the implementation vendor being under contract. The reason for this is:

a. The implementation vendor should influence these derived, technical requirements, based on both their solution approach and their expertise. They may propose better ways to implement something than the business users would have thought of on their own.

b. The business organization won't get the tier-three technical requirements completely right (no-one does). If the contract is based on government developed detailed technical requirements, then any change to these requirements will be a scope change and require a change request.

So overall, our requirement related recommendations are:

1. Define the tier-one and tier-two requirements very well.

2. Use those as the basis of the acquisition.

3. Include a technical requirement definition stage at the start of the SDLC, during which the vendor and PRD will jointly define and agree on the derived technical requirements. These will be specified either electronically or in a Software Requirement Specification and approved during a Software Requirement Review (a gate review). A traceability matrix will trace back and forth to the business requirements, and from there to the strategic requirements. Once those technical requirements are defined and approved, they will be put under configuration management, and they will be the ground truth for building and testing the system.

4. It's fine to provide technical requirement work completed thus far as input to the process, and even to include that work in the bidder's library. But it should be clearly stated that the detailed technical requirements are input to the requirement process, not the final technical requirement specification.

5.1.1 Functional Requirements.

Good requirements fully define the required system functionality, are internally and externally consistent, are verifiable (e.g., through testing or inspection), and are traceable[14]. Functional requirements describe the business functions that the system must perform. They may be defined in a variety of ways and at different levels of detail, including requirement lists (e.g., in spreadsheets), in use-case scenarios, in user stories, in flowcharts, in wireframe diagrams, and so on. Requirements are generally expressed initially as stakeholder requirements, which define the business functionality in a system/implementation independent manner. These requirements are defined by business users or subject matter experts, typically with the assistance of a Business Analyst. Those requirements are then expanded by the technical team with implementation specific details, normally under the leadership of a Systems Analyst.

[14] Per IEEE 830.

Those expanded requirements are termed, “derived requirements.” Traceability matrices are used to ensure backward and forward requirement traceability across all requirement forms.

The CARS functional requirements consist of a Project Approval Lifecycle (PAL) Stage 1 description of the high-level business need[15], roughly 75 use-case scenarios documented in Visio, approximately 30 business process descriptions (training aids) in Word, about 30 “to-be” business process diagrams in Visio, approximately 800 detailed requirements in Azure DevOps (ADO) and Excel, and approximately 1,100 business rules in Excel and ADO. There are also several wireframes and user stories created as part of the most recent CARS related work that may have some utility. Overall CARS received a score of High in this category, as we believe that 95% of the CARS requirements are documented in the above material.

The biggest challenge for CARS going forward will be to organize the existing requirement material. They currently exist in multiple formats and versions across multiple locations. CARS should:

- Review all requirement related material, identify the most correct version of the most useful artifacts, and place that material under configuration and version control. A copy should be included in a bidder’s library.

- Clarify the role of automation with respect to the requirements (especially the business rules). In many cases, it is currently unclear if a given requirement will be fully automated, partially automated, or remain a manual or external process.

- Clearly differentiate between as-is and to-be processes, where as-is process descriptions may be useful for background information, while the to-be processes are the actual requirements to be verified as part of the new system acceptance.

We believe that a significant amount of the existing CARS functional requirement related material is reusable going forward.

5.1.2 Non-Functional Requirements.

Non-functional requirements may be broken down into two major categories:

- Requirements pertaining to the implementation project itself, including things like deliverables, standards, activities, staffing, training, and transition out. The CARS phasing strategy discussed earlier would also fit into this category.

- Non-functional requirements pertaining to the system itself. Institute of Electronic and Electrical and Electronics Engineers (IEEE) Standard 1233 provides examples, including:

[15] CARS - S1BA.pdf

- Reliability.
- Availability.
- Maintainability.
- Performance.
- Accessibility.
- Security.

There are also CARS specific non-functional requirements that should be called out going forward. Specifically:

- The requirement to accept but tag dirty data for later reporting and clean-up. This may include differentiating between errors and warnings in the data and may include a severity level.

- The requirement to track version changes to the field level for most data.

- The requirement to easily change business rules and workflows.

Overall, we assigned CARS a score of High in this category (95% complete). The non-functional requirements documented in the original CARS Request for Offer (RFO) (and the resultant Perspecta contract) are mostly still valid and appear to be relatively comprehensive. In addition, the CARS requirement matrix includes some of the necessary non-functional requirements.

Going forward, we believe that the best source of non-functional requirements is the original RFO that resulted in an award to Perspecta, plus the non-functional requirements in the existing requirement workbook, with a review and update as needed.

5.1.3 Requirement Structure.

When a project has more than a couple of hundred requirements, it's important that those requirements be organized into a structure for purposes of comprehension, scheduling, and review. As a best practice, a three-level hierarchical structure is typically used. The structure should generally align with the business functional areas so that related business functions are designed, implemented, and tested together.

CARS received a score of High in this category. Each of the requirement documents referenced above maps those requirements back to some form of Business Category, creating a two-level hierarchy. For example, the CARS requirement matrix uses the categories shown in Table 5.

Table 5: CARS Requirement Matrix Categories

Requirement Categories
API

Requirement Categories
Availability
Correspondence
Data Migration
Data Retrieval
Filer Disclosures
Filer Registration
Financial Transactions
Global
Interoperability
Localization
Maintainability
Miscellaneous
Payment Processing
Performance
Security
System
System Administration
System Reports
User Account Maintenance
User Experience/Usability
User Notifications
User Support

In reviewing and finalizing the go-forward requirement, consideration should be given to using a standardized three-level hierarchy to group requirements. When developing these groupings, consider structuring them to support the different groups that will be responsible for design, implementation, and testing of the functional and non-functional capabilities. So, for example, the first level might break requirements down by requirement category (e.g., functional, non-functional, security, system, etc.); the second level might break the functional requirements down by business unit and might break the security requirements down by areas such as physical security, access control, intrusion detection, etc. The third level down would then be the individual high-level requirements (e.g., the list of forms, reports, interfaces, workflows, etc.)

5.1.4 Requirement Storage.

Requirements should be assigned a unique Identifier (ID), stored electronically, and managed with both configuration and version control. The unique ID is important to support traceability throughout the life of the requirement (design, implementation, testing, deployment). They should be stored electronically to facilitate tracking and managing. Configuration control is critical to ensure that only the approved version of requirements is used. And version control is helpful to understand the history of the requirement, especially in a situation where implementing a requirement results in unintended consequences.

In the case of the CARS project, the score was average. Requirements were assigned a unique ID and were stored electronically using a combination of Excel/SharePoint and ADO. While both SharePoint and ADO can support configuration control and versioning, we did not see evidence that the SOS has processes in place to control the requirement configuration or track/manage version data.

The existing requirements will form the basis of the requirement rework, as described above.

5.1.5 Traceability.

Requirements should be traceable both backward to their source and forward to derived requirements. Backward traceability is important to fully understand the requirement during implementation and testing, while forward traceability is needed to ensure that no requirements are either forgotten or only partially implemented. Reports (or spreadsheets) should be available and used to ensure full requirement coverage, and traceability should be verified and approved. While Independent Verification and Validation (IV&V) can often support the verification process, the actual approval of traceability should come from the business owners. This is important because only the business owners can correctly determine if the derived requirements really do capture all the required functionality envisioned as part of the source requirement.

In this case we did find significant evidence of requirement traceability. For example, requirements were traced within ADO between stories and the original CARS requirement spreadsheet, business requirements were traced in Excel back to underlying legislative requirements, and test cases were traced to requirements. However, we did not find evidence that all this traceability was verified and approved by the SOS business users.

The requirement traceability data, especially tracing back to the underlying legislative requirements, will be useful when creating the updated requirement hierarchy going forward.

5.1.6 Requirement Status.

Because requirements define the scope of the project, it is important that the status of each requirement be tracked through design, implementation, unit testing, SIT, UAT, and final approval.

In the case of the CARS project, we assigned a score of Very High because requirements were actively tracked and managed from design through testing.[16] ADO was the primary method of tracking requirements through this process.

[16] RLS Stats from RLS Mapping Sheet.pptx.

The existing requirement status information is not usable going forward, but some of the processes used (e.g., tracking requirement status within ADO) is usable.

5.1.7 Requirement Management Processes.

Because effective requirement management is so important to the success of the project, processes should be in place for managing requirements and controlling changes to requirements. Those processes should be documented, and they should be consistently followed.

We assigned CARS a score of Average in this category. The project did have an approved change control plan that addresses many of the requirement management processes.[17] There was also a less formal set of processes covering smaller changes.[18] Overall, it appears that requirement related processes were followed but not consistently. This was especially true with respect to the specific scope of what was expected with respect to UAT, where our interviews highlighted the significant disconnect between requirement related expectations of the technical stakeholders (e.g., OSaaS/Project Management Office (PMO)/Information Technology Division (ITD), the business stakeholders (i.e., PRD), and the external stakeholders (e.g., Netfile)).

We recommend that a new requirement management process be implemented as part of the requirement review process recommended above. This requirement management process should include, among other things, requirement related governance.

We do not believe that there are any reusable artifacts in the existing requirement management material.

5.1.8 Business Process Improvement.

A major automation project should do more than simply automate the existing, as-is business processes. One of the key value propositions of any new automation project is the opportunity to use automation to do existing things better, or to do new things that are not possible now. Those business process improvement opportunities must be defined early in the system lifecycle so that they may be incorporated into the requirements. Understanding these opportunities will often drive fundamental design and architecture decisions as early as the acquisition stage and may point the way to process improvements even before system design and implementation.

We evaluated the CARS project as High, in that many potential opportunities for improvement are incorporated into the existing requirement related artifacts described above. However, we believe that there is an opportunity during the requirement refresh to go even farther in terms of potential opportunities for improving the way the PRD business processes work. For example, we believe that by allowing the system to accept dirty data that is automatically tagged with error or warning conditions,

[17] CARS Change Control Plan V2.0 Approved.pdf.
[1818] Documented in “Sys CR Process.docx”.

automated processes may be put in place to support the cleanup of that data through outreach to the data submitter, through automated processes, or through work by PRD analysts.

The currently identified opportunities for business process improvement (in other words, the existing to-be system requirements), as documented in the existing requirement artifacts, should be mostly reusable going forward.

5.2 Project schedule management.

Project schedule management involves more than simply creating and updating Gantt charts. Effective project schedule management means that:

- The generated schedule is comprehensive, capturing the work required by all stakeholders. We term this Integrated Project Scheduling, as differentiated from the project schedules (often more detailed) that may be created by specific teams for internal use.

- The schedule must be realistically achievable, which involves accurate estimating and ensuring that the project resources are not overloaded.

- The schedule must be monitored to identify progress against plan. This monitoring includes both week-by-week schedule assessments plus more formal milestone-based assessments of actual project status versus expectations.

- Finally, the schedule must be revised as needed based on actual results. These revisions should be proactive based on early predictors of problems, rather than last minute reactive changes. Significant updates must be both approved through a governance process and communicated to all affected stakeholders.

Figure 12 shows the CARS Project Schedule Management Assessment scores in each of the relevant eMRI KPAs, using a scale of 1 (very poor) to 5 (exceptional), with 3 as typical or average. The biggest issue for CARS was not with the project's on-going schedule management, but rather with poor estimating and resourcing, resulting in an unachievable schedule, followed by inadequate milestone reviews to catch the problem early.

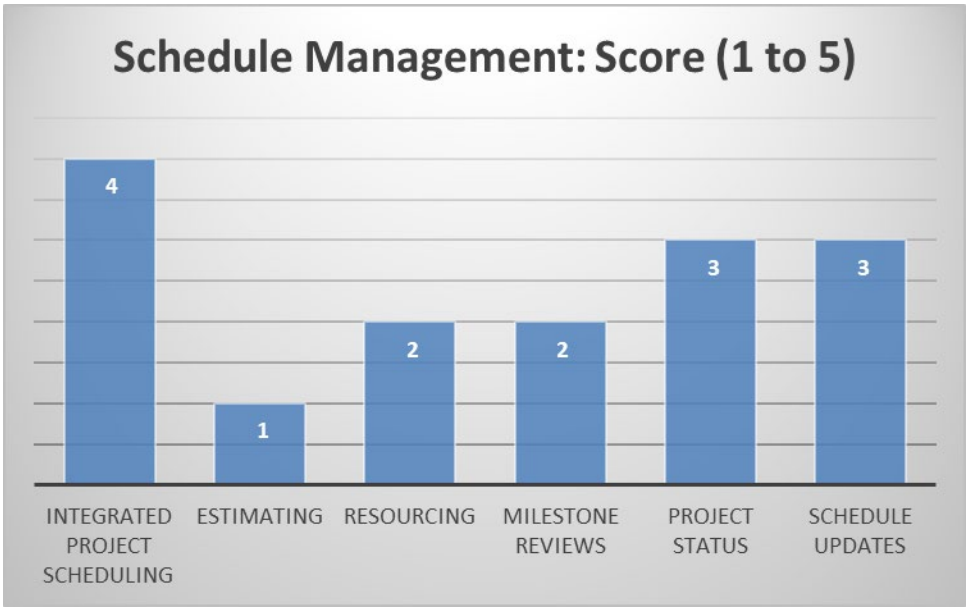

Figure 12: CARS Project Schedule Management Assessment

In the remainder of this section, we'll cover each KPA in more detail.

5.2.1 Integrated Project Scheduling.

Integrated project scheduling involves creating, communicating, and maintaining an integrated schedule covering the major activities, deliverables, resource requirements, and milestones for all project stakeholders, both internal and external. For larger projects, the Integrated Project Schedule (IPS) will be supported by more detailed, stakeholder specific schedules that are developed and maintained by those groups. So, the IPS must achieve a balance of detail, with enough detail to effectively monitor and control the project, but not so much detail that the workgroups are micro-managed. Excellence in integrated project scheduling (a score of very high in eMRI) implies that an IPS exists that captures all the required project work, assigns work to individuals or groups, and includes task dependencies.

For the CARS project, the assigned score was High. An IPS in Microsoft Project does exist and was updated (OSaaS deliverable I.3). The IPS does include the OSaaS contractual deliverables, the software development sprints, deployment activities, and activities necessary to transition to maintenance and operations (M&O). However, the IPS was completely OSaaS centric. In that sense it had both too much detail and too little detail—too much detail for the OSaaS activities, and either no or inadequate detail for other stakeholders such as the external interface partners, PRD, and ITD. Some of these stakeholder schedule expectations does exist in other places (for example, in high level Gantt charts presented in PowerPoint presentations), but the point of an IPS is that it serves as the scheduling system of record for the entire project (all stakeholders), and in that sense the existing IPS is somewhat lacking.

However, we did observe that there appeared to be significant project scheduling issues surrounding the approach to the SDLC. The implementation vendor was attempting to

follow a modified Agile approach, and this created problems for the PRD staff, which were more comfortable with more of a traditional, waterfall approach. In general, we are not a fan of pure Agile development for government projects, as the government projects are driven by fixed budgetary and requirement constraints. Instead, we recommend a generally waterfall, milestone driven approach, but with Agile techniques used to finalize the derived requirements and user interface design for components of the system. In this way, the system design is incrementally finalized in a manner similar to rolling-wave project management as described in the PMBOK.

While the IPS is not reusable going forward, the *CARS Project Management Plan*, OSaaS deliverable 1.2, does have reuse potential going forward with an update/revision cycle.

5.2.2 Estimating.

Estimating involves estimating the resources and time required to perform the various project activities. There are a variety of approaches to estimation that each have advantages and disadvantages, and that are appropriate in specific circumstances. These are:

- Catalog look-up: Look up the price in a standard catalog such as a California Multiple Award Schedule (CMAS) schedule or manufacturer's catalog. This primarily applies to the purchase of commodities.

- Learning curve: Calculate costs based on historic costs adjusted for an exponentially shaped learning/efficiency curve. This primarily applies to manufacturing.

- Analogy: Calculate costs by comparison to previous similar projects, perhaps with project specific adjustments.

- Parametric: Parametric estimates may be high-level or detailed. High level parametric estimates forecast costs based on a parameter that is not directly related to the mission or business function of the item, but that is predictive of cost/effort. For example, the number of square feet is often used as a high-level parametric variable when estimating construction costs. Detailed parametric estimates use a High-Level-Object (HLO) catalog of items that have a historic relationship with cost/effort; but which also have a direct relationship with the mission or business functionality. HLO catalogs allow benchmark-based estimation. For some Information Technology (IT) HLO Catalog samples, see Table 6.

- Bottom Up: Unlike detailed parametric estimation, where the estimation components are specific deliverable capabilities or things, bottom-up estimation begins by defining the project in terms of work activities to be performed (e.g., code, unit test, project management, detailed design). Individuals familiar with

the work to be performed will then directly estimate their time based on personal historic experience and an understanding of the requirements.

Table 6: Sample IT HLO Object Catalogs

Agile	IFPUG_FP	ExcelerSize		
Product Suite	EI	APPLICATION	DATA WAREHOUSE	SUPPORT
Product	EO	Batch	Cube/OLAP Datamart	Code Cleanup
Theme	EQ	Business Requirement	ETL	Configuration
Epic	ILF	Extension	Portal	Defect Repair
Story	EIF	Form-Inbound	Predictive model	Incident
		Form-Outbound	Universes	Minor Enhancement
UML	IVR	Interface		Patch
Actors	Admin Screen	Page	PROJECT LEVEL	RA Uplift
Use-Cases	Call Initiation	Report	Consulting-Configuration	Support-Analysis
Class-Control	Call Tree Option	Service	Consulting-Performance	Support-Application
Class-Interface	Interface	Table	Consulting-Security	Support-Database
Class-Other	Report	Workflow	Consulting-Other	Support-Integration
Tables	Security Profile		COTS-Application	Support-Other
Methods	Table	DATA CONVERSION	COTS-Module	Support-Security
	Voice Message	Convert-Database	COTS-Component	Support-Tech Writing
		Convert-Table	Framework	
		Convert-Field		
		Convert-MREC		

In general, software projects such as CARS should use a combination of detailed parametric estimation using benchmark data and bottom-up estimation. To help avoid bias in bottom-up estimates, techniques such as Program Evaluation and Review Technique (PERT) or Delphi may be used. PERT does the estimates in terms of best case, expected case, and worst-case estimates, while Delphi uses a consensus of experts approach.

For the CARS project, the score was very low. The project estimates were wildly off both in terms of resources and schedule. Estimates seem to have been prepared based on a “What do we want it to be” estimation approach. There was some attempt to use more formal estimation techniques to estimate the UAT specific effort[19], to approximate the effort by specific SMEs[20], and a catalog lookup approach was used for many Other Direct Charge (ODC) items such as software licenses.

We do not believe that any of the existing estimation related material is reusable going forward.

5.2.3 Resourcing.

[19] See for example, CARS UAT Effort Estimates v01.xlsx.
[20] CARS Project PRD Resource Estimates.xlsx.

Resourcing involves determining resource requirements by period (day, week, or month); comparing those requirements to resource availability; and adjusting the project activities to achieve an acceptable level of resource usage. Resources may be individuals, labor categories, organizations, or in some rare cases, ODC items such as equipment or facilities. The process of adjusting activity durations or schedule to meet resource constraints is termed resource leveling. Other approaches to leveling resources include adding additional resources or reassigning resources between the activities. The goal is to adjust the project schedule and resources so that no resources are overloaded during the project, and the project has adequate resources available at each point in time.

Note that for staff that are not assigned to the project fulltime, the resource leveling must be performed based on their available hours. So, if a SME is only available to the project for 10 hours per week, then 100% utilization for that individual would be 10 hours per week.

One common misconception is that resource overloading is acceptable on the assumption that personnel will be willing to work overtime. While this strategy will work for limited situations (e.g., short duration projects, critical but short activities), it is high risk to use this approach as a long-term strategy. The reasons for this are:

1. There is uncertainty in all activity estimates. If staff are leveled appropriately, then tasks that take longer than expected can sometimes be completed on schedule by working some extra hours. This built-in recovery flexibility is removed if staff are already overloaded.
2. Long term overloading of staff will result in burn-out and higher than normal staff turnover. This high staff turnover, and the resultant learning curve for new staff, will tend to exacerbate schedule problems.
3. Overloaded staff will tend to perform lower quality work and introduce more defects into the system. This will often show up as problems during SIT and UAT, where the system will have higher than expected defect rates.

In the case of the CARS project, the IPS did not include resource allocations at all, so no attempt at resource leveling within Microsoft Project was apparently attempted. During the interviews, we were repeatedly told that staff were overloaded and asked to work long hours. This included OSaaS staff, PRD staff, and external stakeholders. There was some limited attempt to do resource leveling for PRD staff[21], and when scheduling UAT there was some attempt to manage the work versus the availability of SMEs[22], but it appears that this work was more reactive in nature than pro-active. In other words, staff availability had become an issue and there were attempts to manage that availability. Overall, we assigned CARS a score in this category of Low.

We do not believe that any resource leveling related artifacts are useful going forward.

[21] CARS Project PRD Resource Estimates.xlsx.
[22] SOS Team PRD & Test Team Testing Resource Calendar.xlsx.

5.2.4 Milestone Reviews.

Milestone reviews are formal reviews during which the project status as of that review is analyzed. They are important because they allow the project and oversight teams to formally assess project progress, allowing early corrective action if necessary. The most effective milestone reviews are gate reviews, where successfully passing an agreed to performance threshold is a criterion for moving forward to the next phase of development. Because a primary intent of the milestone reviews is to allow early corrective action when needed, they should be conducted at most every six-months or 25% of the project duration, whichever is less.

Note that while successfully completing a milestone review is always a milestone in the project plan, not all milestones in the plan are milestone reviews.

While the specific reviews to be conducted may vary based on the selected SDLC, the following reviews are common[23]:

- Software Requirement Review, during which the technical requirements are confirmed and approved.
- Detailed Design Review, during which the specific architecture and design is finalized.
- Test Readiness Review to confirm readiness for the start of SIT and UAT.
- Production Readiness Review to confirm readiness for deployment and cutover.
- Post Implementation Evaluation Review to confirm that the system is operating effectively, meeting intended objectives, and ready for transition to M&O.

In the case of the CARS project, we assessed the project at Low. There were some identified milestone reviews toward the end of the project, primarily related to UAT and deployment, however these were not formal reviews. And with no reviews until the end, project problems were not exposed until late in the SDLC execution.

We do not believe that any milestone review related artifacts are useful going forward.

5.2.5 Project Status.

Effective management of project status can be an early and reliable predictor of project health. This includes the obvious predictive value with respect to schedule, but it will also typically be an effective predictor of problems with budget, and often it will predict problems with quality.

[23] For more on this, see IEEE-12207.

Exemplary organizations will baseline the project plan, then capture actual progress, financial expenditures, and hours worked throughout the project life. Actual project status and expenditures are used to compute earned value metrics, including the Schedule Performance Indicator (SPI), Cost Performance Indicator (CPI), Estimate to Complete, and Estimate at Complete. Trends over time are tracked to determine if project health is improving or deteriorating. Results are provided to management in the form of dashboards.

CARS was assessed as Average in this dimension. While we did not see actuals consistently captured in the project schedule, actual status was captured using spreadsheets and Gantt charts[24], expenditures versus budget were tracked and managed[25], and weekly status reports were prepared that outlined the current percent complete for major project phases.

We did not find any project status related artifacts that are useful going forward.

5.2.6 Schedule Updates.

It's safe to say that schedules for major information technology projects will always require at least some updates. Reasons include project scope changes, resource availability changes, schedule slippages, and required risk mitigation or response. Exemplary organizations are characterized by:

- The schedule is kept accurate through regular updates.
- The updates are approved through a formal governance process.
- A baseline (original) schedule is maintained, and changes against that baseline are noted.
- Trends in areas such as project scope are tracked and reported.

We assessed CARS as Average, in that the project did update the project schedule on a regular basis, however those updates were not consistently managed through a formal governance process, and while the integrated project schedule was baselined, we did not see evidence of using that baseline information to report on trends and to highlight differences.

We do not see any schedule update related artifacts that will be useful going forward.

5.3 Risk management.

[24] For example, see 2020 0729 SOS CARS Project - Gantt Chart.pdf.
[25] For example, see 08 SEP 2021 CARS ESC Budget Update.

To be effective, risk management should be integrated into all aspects of the project. Formal risk management is a discipline that goes well beyond the informal process of "watching out for bad things." It's helpful to begin with a rigorous approach to some risk related definitions, as there is often inconsistency in the meaning of risk related terms. The following terms are defined in ISO/FDIS 31000:2009:

- **Risk:** effect of uncertainty on objectives.
- **Level of Risk:** magnitude of a risk, expressed in terms of the combination of consequences and their likelihood.
- **Consequence:** outcome of an event affecting objectives.
- **Likelihood:** chance of something happening.
- **Event:** occurrence or change of a particular set of circumstances.
- **Risk Assessment:** overall process of risk identification, risk analysis and risk evaluation.
- **Risk Management:** coordinated activities to direct and control an organization with regard to risk.
- **Risk Treatment:** process to modify risk (e.g., avoidance, mitigation, acceptance, transfer).
- **Residual Risk:** risk remaining after risk treatment.

One key aspect of formal risk management is that risks are any area of uncertainty, so risks include both threats and opportunities. In other words, some areas of uncertainty might reduce costs, improve performance, decrease schedule, and so on. These items should be managed through the formal risk management process as well. So, if a risk register only contains threats (bad things), it might be useful to examine the risk management process to determine if the project is missing risk opportunities.

Figure 13 shows how the various risk related KPAs typically fit together. Executive interviews and governance documents such as the Project Charter are used to establish a risk context, which would include organizational risk tolerance and risk related project boundaries, objectives, assumptions, and constraints. A variety of interviews, documentation, and benchmark/historic data is then used as input to the risk assessment process. Risk assessment includes risk identification, analysis, evaluation, and the development of a mitigation strategy. Risk evaluation is often a two step-process. First, risks are evaluated using an ordinal scale (e.g., Very High to Very Low), and the results of that process are used to prioritize risks for more detailed evaluation. The more detailed evaluation involves attempting to place specific numbers (time, money) on the risk, and may involve techniques such as Monte Carlo simulation. Finally, for the risks where the threat or opportunity is significant, a risk treatment plan is developed.

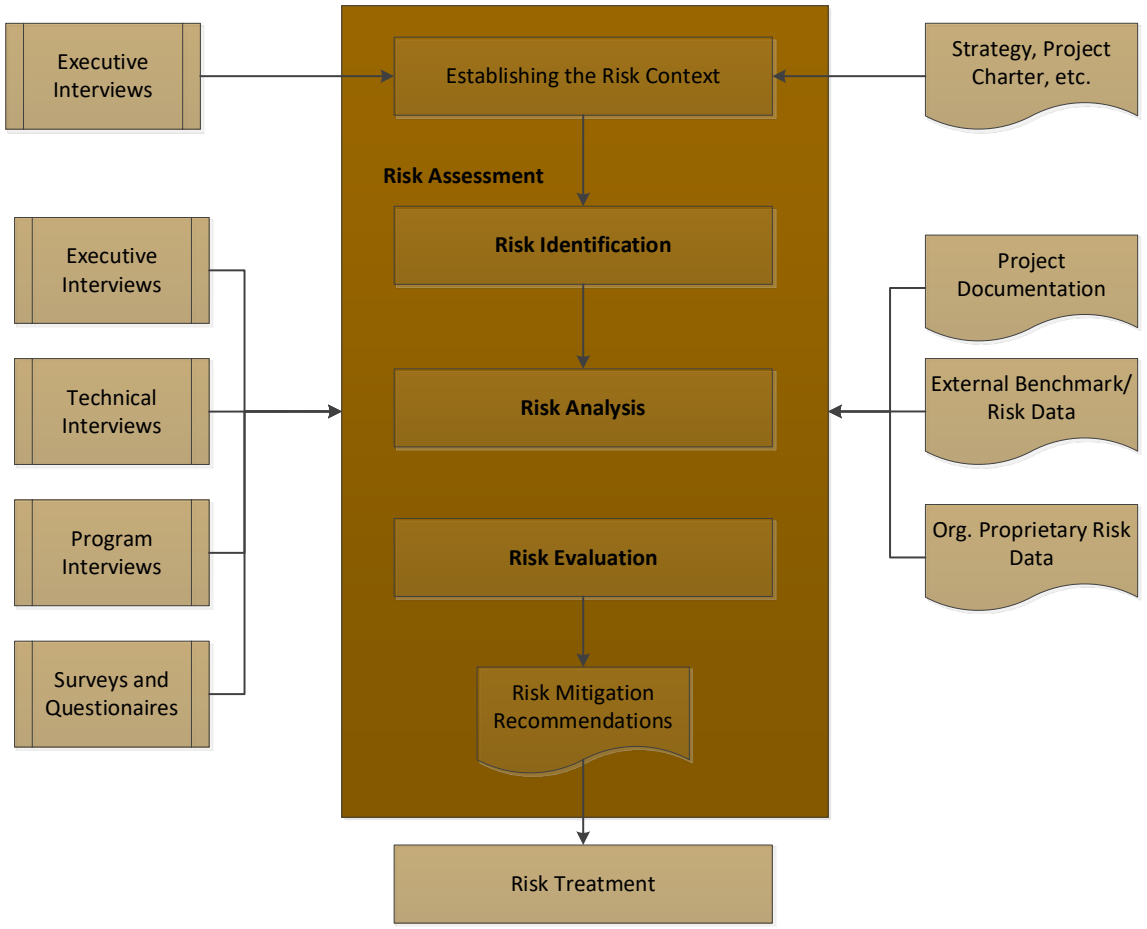

Figure 13: A risk management framework.

Figure 14 shows the CARS scores across each of the eMRI risk management KPAs. The biggest problem with CARS risk management was not with respect to the processes themselves, but with the project's reluctance to accept the validity of extremely high threats to the project goals and objectives. For example, during the interviews we were told that in several cases PRD input to the risk process was rejected, and no risk items were opened, because the PRD input did not describe valid risks. But in reviewing the PRD emails in question, it was apparent that PRD was attempting to raise the very risks that ultimately were at the heart of the failure of CARS to have a successful UAT, and the subsequent pause and restart of the project. In other words, the CARS risk managers did not accept those PRD generated risks because they went contrary to their understanding of the current project status.

Figure 14: CARS Risk Management Scores

In the following sections we'll address each of these KPAs in more detail.

5.3.1 Risk Planning.

For exceptional organizations, a Risk Management Plan (RMP) will be developed and approved, plus that RMP will include suitable budgets for risk related activities, including risk mitigation activities. For example, suppose that a risk is identified related to the interfaces to external filing partners. The risk mitigation strategy might include:

1. Regular interface control working group meetings with those partners.

2. Putting in place a Memorandum of Understanding (MOU) with each external partner, including joint testing related responsibilities.

3. Development of a test harness that the project can use to test the interface independent of the external vendors.

4. Development a second test harness that the vendors can use to test the interface independent of CARS.

All four of these risk mitigation activities have a cost associated with them, so they would all require a budget, and they would normally be managed by the risk team.

CARS received a High evaluation for this KPA. An approved Risk Management Plan does exist[26], but we did not find evidence of budgeting or managing risk mitigation activities.

[26] CARS Risk Management Plan V2.0 Approved.pdf.

We believe that the Risk Management Plan (RMP) would be reusable with modification going forward.

5.3.2 Risk Identification.

As previously discussed, the risk identification process should include both threats and opportunities. For example, suppose that OSaaS is planning to do all the data conversion work from scratch using the CAL-ACCESS database. There might be an opportunity to reuse previous data conversion rules or cleaned data from the Perspecta effort. Based on the potential cost savings, it might be worth investing the exploratory work to analyze the feasibility of this. This line item should then appear in the risk register.

Risk identification should be integral to all levels of the project, including all project meetings, plus it should be a specific agenda item for status meetings. Risks should come from all stakeholders, both internal and external. Identified risks should be assigned a unique ID and tracked to resolution in a risk register.

In the case of the CARS project, we assigned a score of Average. We did find evidence of risk discussions incorporated in the on-going project status reporting and meetings. We did find a risk register, maintained in ADO. The identified risks were limited to threats, so the project was deficient in not also looking at opportunities. During the interview process we also found that risks were not actively solicited from external stakeholders; risks from internal business stakeholders were incorrectly filtered as discussed above, and that there was too much emphasis placed on the form of the risk submission, rather than the substance. For example, in one interview we were told that risks were not accepted because they were submitted as emails or spreadsheets rather than through the formal risk process using specific forms. While forms are useful and should be encouraged, the ultimate goal of risk management is to be as comprehensive as possible, not simply to have the forms completely correctly.

We do not believe that there are any reusable components in this KPA.

5.3.3 Risk Categorization.

Risks should be categorized by both the risk source/type (e.g., business/financial; health and safety; legal; project; quality; security; technical; other or multiple) and the impact category (impact to the project, impact to the organization, impact to external stakeholders). This categorization is useful in terms of assessing the likely risk exposure and accessing relevant risk related benchmark data. It will also determine stakeholders that should be involved (either directly or in the form of reports) with the risk related activities and status.

CARS scored Very Low in this KPA; in that we did not see any indication that CARS risks were categorized in any way.

We do not believe that there are any reusable components in this KPA.

5.3.4 Risk Assessments.

As previously discussed, risks should initially be assigned a qualitative probability (very low to very high), consequence (very low to very high), and overall exposure (very low to very high). This qualitative analysis can then be used to prioritize risks, so that quantitative approaches can be used for risks with the greatest exposure. Exceptional organizations will perform that quantitative analysis of risks using techniques such as Monte Carlo simulation.

In the case of the CARS project, the assessed score was Low. The organization used a single magnitude factor to categorize risk exposure (Very Low to Very High). Skipping the step of separately looking at probability and consequence is both less accurate and loses some valuable insight in terms of developing a mitigation strategy. As a minimum, organizations should separately look at probability and consequence, with exposure then calculated based on these two values.

We do not believe that there are any reusable components in this KPA.

5.3.5 Risk Management.

For risks with high exposure and for significant opportunities, a risk management strategy should be defined and implemented to decrease the risk exposure for threats and increase the potential value of opportunities. Where risks remain high following mitigation, an approved contingency plan should be in place.

In the case of the CARS project, we assigned a score of Low to this KPA. In reviewing the project weekly status reports, it appears that the CARS risk management approach primarily involved monitoring and responding.

We do not believe that there are any reusable components in this KPA.

5.3.6 Risk Monitoring.

As part of risk monitoring, risk probabilities and impacts should be updated on a regular basis. Trigger events/criteria for risks should be identified. These are used to clearly define the point where a risk becomes an issue. Risk monitoring should identify which risks have become issues as part of regularly scheduled status meetings. Notification procedures should be in place to notify relevant personnel, including the Project Sponsor, of risks that become issues or otherwise have significant changes.

CARS received an assessed score of Average. We saw evidence of risk updates as part of the weekly status report, and an issue log was maintained. We did not see evidence that trigger events or criteria were published or identified. While changes in risk status was discussed during status meetings, we did not find evidence of a more

formal process of identifying the various stakeholders that needed to be notified in the event of a change in risk status, and then notifying those stakeholders when a risk status changed.

We do not believe that there are any reusable components in this KPA.

6 CARS Process Assessment

In this chapter, we cover the CARS process assessment composed of the following five KPA categories of the eMRI model:

1. Contract management and vendor negotiations.
2. Communications management.
3. Governance and sponsorship.
4. Organizational change management.
5. Quality management.

Within each KPA, CARS was assessed with a score of 1 to 5, where 1 would be highly ineffective, 3 would be moderately effective (average or typical), and 5 would be highly effective. The results are shown in Figure 15.

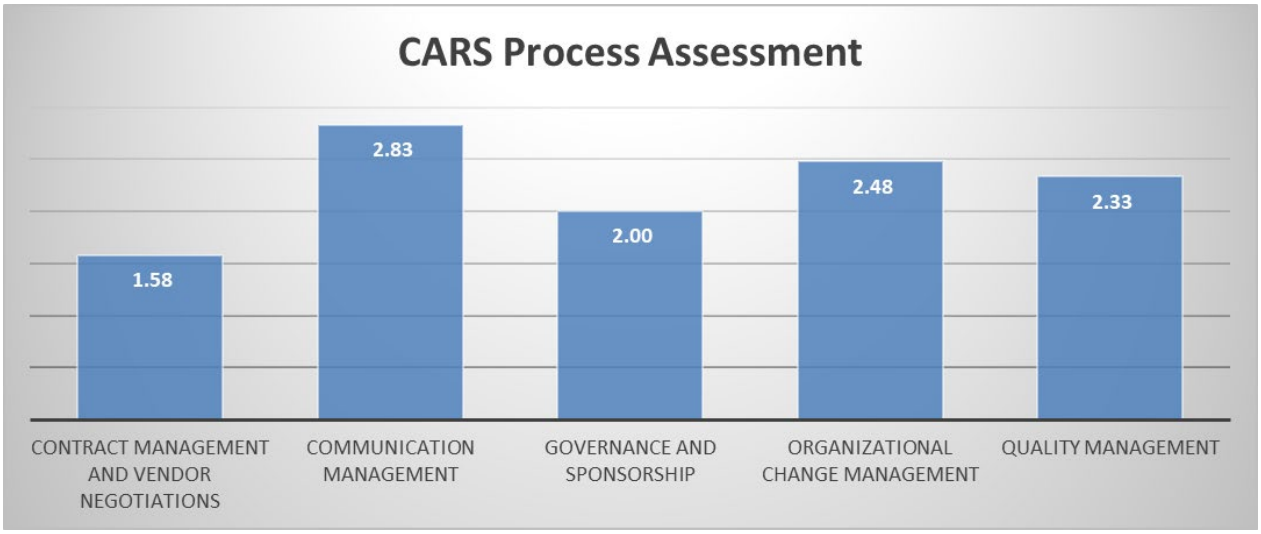

Figure 15: CARS process assessment results.

In the remainder of this section, we'll cover each KPA category in more detail.

6.1 Contract Management and Vendor Negotiations.

In this section we cover contract management and vendor negotiations. Because most large government IT projects are at least partially dependent on contractors for execution, this area is pivotal for project success and requires carefully developed strategy to increase the value of vendor (or contracted) services. These processes manage the creation, implementation, and evaluation of contracts to maximize project performance, solution quality and minimize overall project risk.

Contracts, being legally binding, determine vendor accountability, project clarity, vendor-state relationships, pricing structures, the scope of work, rights and obligations, timelines for projects, and warranty provisions. Given how critical contracts are, effective contract management can dramatically improve the performance of the project.

Similarly, problems in this area can significantly reduce the probability of project success.

The State's contract management team, including administrative and technical roles, need the appropriate level of experience contractually managing mission-critical enterprise technology projects, including training when processes such as WOAs are introduced. Trained government contract managers should be involved in the project from acquisition planning through project close-out, ensuring that State contracting regulations and guidelines are followed.

Our assessed score for CARS in this category is shown in Figure 16. The CARS project has performed a "lessons learned" analysis that we have found consistent with our findings. We recommend this analysis be utilized during the recasting of the CARS project.

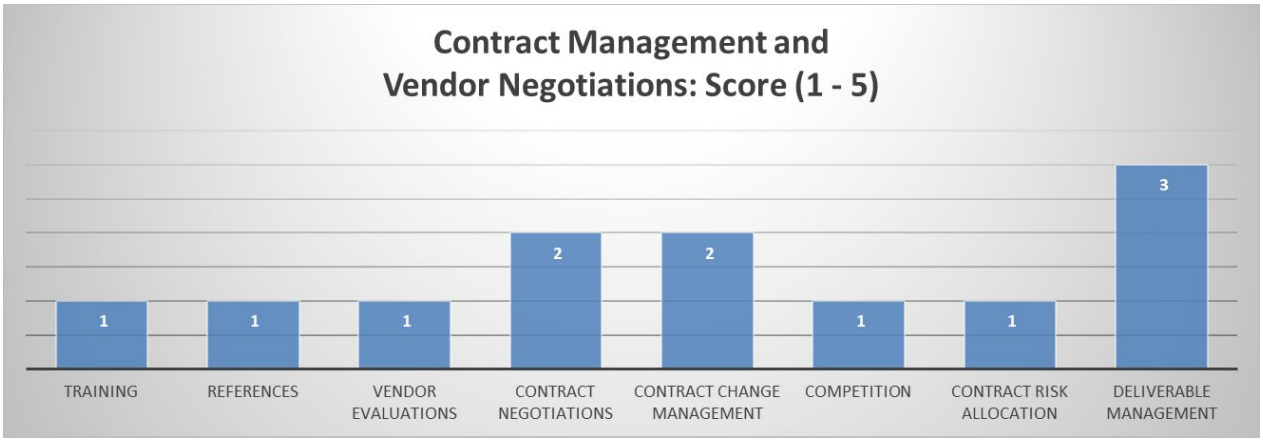

Figure 16: CARS contract management and vendor negotiations scores.

6.1.1 Training.

The contract management staff should receive suitable contract management training, including annual refresher training, include training in the specific contract types used for a given project (WOA based, in this case). This training should cover all aspects of the acquisition and support for the contract, including regulatory requirements for competitive acquisitions, development, or review of the statement of work, deliverable definition and acceptance, contract change management, and so on.

In the case of the CARS project, we were told during the interview process that the assigned Contract Manager did not understand the WOA process, that he asked for training or assistance in this area, and that none was provided. We also did not find procedures and contract officer training records in general covering complex enterprise IT contracts such as CARS.

We did not identify any contract officer training related artifacts that are useful going forward.

6.1.2 References.

Past performance of a vendor is an important consideration for the procurement of an enterprise system. In the case of the CARS project, we confirmed that reference checks for the selected system integrator (OSaaS) were not conducted.

We did not identify any reference related artifacts that are useful going forward.

6.1.3 Vendor Evaluations.

A successful vendor evaluation practice begins with well-organized procurement planning aligned to the organizations vision, mission, strategy, and project charter. The project planning team's ability to develop clear, complete requirements, project scope, and service level agreements leads to fair and objective vendor evaluations for contract award and continues through to contract performance. Best practices for vendor management include objective, documented contractor evaluations that start during the bid process. Using that as a precedent, ongoing evaluations should be completed throughout the project life cycle (e.g., milestones, deliverables, contract closeout) to support contract management.

In the case of the CARS project, there was little evidence that a formalized vendor evaluation framework (methods or artifacts) was leveraged, due to the nature of the procurement.

We did not identify any vendor evaluation related artifacts that are useful going forward.

6.1.4 Contract Negotiations.

A state-side team experienced and informed in enterprise system contract negotiations is critical to overall project success. The State team should be equipped with well-organized, quantified project scope that is properly chartered, and supported with independent government cost estimates to assist in fair and reasonable negotiation with each vendor. Contract negotiation support is necessary from initial contract through implementation, during change management, and even during maintenance and operations.

In the case of the CARS project, the OSaaS procurement was awarded with no effective competition and managed on a series of Work Order Authorizations (WOAs) between July 2020 and June 2021. With the passive involvement of the contracting office and little to no visibility into the scope of the project, it was difficult to negotiate each WOA effectively.

We did not identify any contract negotiation related artifacts that are useful going forward.

6.1.5 Contract Change Management.

Contract changes are managed in a life cycle that reviews all requested changes to determine if a contractual change is required or if an alternative approach is possible. The ability for the contract manager to effectively collaborate with the appropriate multi-disciplinary team (e.g., legal, project, business, technical) necessary to assess the change is imperative. Because this is a difficult practice to master, many state agencies fall short and fall victim to change order churn. When change is determined to be required, all contractual changes should be processed through an established governance process and approved by the government contracting officer in writing, with no expenditures prior to written approval.

In the case of the CARS project, contract change was difficult to manage due to many factors including the lack of clarity in project scope coupled with the time and material-centric (T&M) Work Order Authorization (WOA) contracting approach.

We did not identify any contract change management related artifacts that are useful going forward.

6.1.6 Competition.

Government policy holds that the best interest of the government is served when contracts are awarded competitively whenever possible. Procurements should be structured to maximize full and open competition, generally resulting in a minimum of three bidders.

In the case of the CARS project, both acquisitions were structured in a way that resulted in limited effective competition:

- The initial CARS procurement included onerous requirements that limited market participation, including California based small businesses with demonstrated expertise in this application domain. Specifically, a mandatory requirement for, “Audited financial statements or SEC 10K filings (including a balance sheet) that support average annual gross revenue of $150,000,000 or more for each of the company’s last three fiscal years.”
- The second procurement was awarded to OSaaS on a de facto sole source basis, in that the procurement was awarded using a limited source competition, the time for vendor response was too short to allow for full and open competition, and incomplete specifications were made available to bidders. The result was a single bidder.

We did not identify any competitive or market research related artifacts that are useful going forward.

6.1.7 Contract Risk Management.

The contract risk management function is intended to assign responsibility (performance, financial or otherwise) to the party best able to control the relevant risk factors. The contract management team needs established policies, process, standards, and principles to support the governance and management of contract risk. With enterprise technology projects, the most effective risk allocation method is to contractually move the risk to the party with the most (perceived) ability to control that risk.

In the case of the CARS project, virtually all the project risk was allocated to SOS by the contract structure (T&M) with minimal ability to control the value received for those hours. Risk identification, mitigation and communication did not appear to be operationalized within SOS for the CARS Project. With the vendor having primary control over the implementation work, risk and the accountability for project risks should have been allocated to them using a firm fixed price, deliverable based contract.

We did not identify any contract risk management related artifacts that are useful going forward.

6.1.8 Deliverable Management.

The contract deliverable management function involved the policies, procedures and practices covering the acceptance of contract deliverables. On IT contracts, deliverables are typically reviewed and approved by both a technical individual and the Contract Officer.

In the case of CARS, we saw evidence of contractors appearing to approve the deliverables of other contractors, which would generally not be appropriate. The appropriate level of deliverable management practices was not established and there was inadequate accountability structure (as found in a R-A-C-I Matrix).

We do not believe that there are any reusable artifacts in this area.

6.2 Communication Management.

In this section we cover Communications Management. The Communications Management function serves the information needs of the project sponsors, management, staff, vendors, and stakeholders. Most project risk, issues, and ultimately success or failure are directly tied to communications. Figure 17 shows the CARS score in each KPA in this category, each of which will be covered in more detail below.

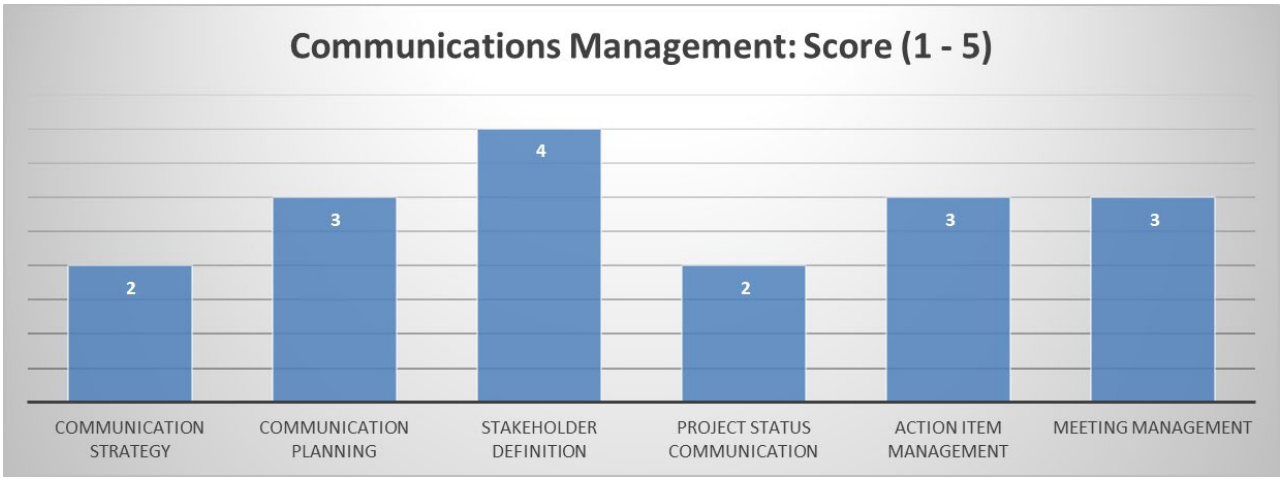

Figure 17: CARS communications management scores.

6.2.1 Communications Strategy

Communication strategy is the means through which an organization plans to use to achieve both communication goals and business outcomes. Communications Strategies categorize the tasks that increase the likelihood that the information needs of the project and its stakeholders are met.

In the case of the CARS project, the project utilized and refreshed the Communications and Stakeholder Engagement (CASE) Plan in the summer of 2020. The purpose of the Plan was to describe, at a high level, the communications and engagement approach, objectives, standards, and activities related to the CARS project's communications and stakeholder engagement. While the plan is seemingly sound, the project had difficulty seamlessly executing it as a strategy across the project.

The CARS CASE Plan and related artifacts will be instrumental in the recasting of the CARS project. This information should be used to develop the new communications strategy for the next generation CARS project.

6.2.2 Communication Planning.

This area includes the collaborative and inclusive activities that create the guidelines, content, and roadmap for exercising the organization's communication strategy. These are critical for conveying important information to target audiences, at the right time, in the right sequence and at the right level of detail. The plan should address the information to be shared, who is responsible for sharing the information, the approach to be use, the frequency, and with whom the information should be shared (target audience).

In the case of the CARS project, the project utilized and refreshed the Communications and Stakeholder Engagement (CASE) Plan in the summer of 2020.

We believe that the CASE plan has some reusable components to be calibrated with the updated CARS communication strategy.

6.2.3 Stakeholder Definition.

Stakeholder definition requires the identification of primary, secondary, and key people or groups impacted by the project or initiative. The Stakeholder definition is valuable for governance, communications and organization change activities.

In the case of the CARS project, stakeholder definitions are documented and were known by the project. However, a stakeholder management strategy was not formally applied to the project therefore execution was ad hoc. Stakeholder information was not widely shared and appropriately matured through the life of the project.

The existing CARS stakeholder list should be useful going forward and should be calibrated with the updated CARS communication strategy.
.

6.2.4 Project Status Communication.

Project status communication serves as open channel of information sharing that occurs at a predefined and predictable interval. Status information is offered through routine publications, posts, and messages, as well as presented at standing governance and informational meetings. A project health dashboard and vendor performance balanced scorecard should be developed and updated on a regular basis. A best practice is to create a project website where current project status information and announcements are available to internal and external stakeholders. Written project status reports are submitted on consistent intervals, and accurately and comprehensively represent the current reality. Project status review meetings are held regularly with project leadership, business, and project team.

In the case of the CARS project, the project managed project status communication according to plan and supported by the Communications Toolkit.

The communication toolkit and related artifacts will be instrumental in the recasting of the CARS project. As with other communication-centric functions, it will be important for project status communications to be calibrated with the updated CARS communication strategy.

6.2.5 Action Item Management.

Action Item Management is the lifecycle that includes discovery, identification, follow through, and reporting of action items. Action items are activities requiring follow up, that are assigned to an individual or a specific group of people. Action items more

granular than standard “task” and typically address blockers or prerequisites for typical tasks or activities.

To improve Action Item Management, a project action item register must be discoverable, available, used, and kept current. Action items are assigned to specific person or group with due dates and status.

In the case of the CARS project, action items were managed, however it was difficult to gauge performance of the function because of the foundational project flaws (e.g., architecture, scope, time, and resource constraints).

We anticipate the action item management function will be reusable in the recasting of the CARS project. As with other communication-centric functions, it will be important for action item management to be calibrated with the updated CARS communication strategy.

6.2.6 Meeting Management.

Meeting management includes the governance of meeting frequency, content, attendee appropriateness, as well as improvement efforts to ensure that meetings are necessary, effective, and as efficient as possible. Meeting management develops and maintains a meeting framework, that includes meeting purpose, attendance, agenda, and minutes. Action items, issues, and risks that are identified during meetings are documented both in the meetings and in the action item and risk registers.

In the case of the CARS project, meetings were managed, however it was difficult to gauge performance of the function because of the foundational project flaws (e.g., architecture, scope, time, and resource constraints).

We anticipate the meeting management function will be reusable in the recasting of the CARS project. As with other communication-centric functions, it will be important for meeting management to be calibrated with the updated CARS communication strategy.

6.3 Governance and Sponsorship.

Enterprise Governance and Sponsorship needs little introduction as it is widely and consistently defined across the globe. As defined by the Information Systems Audit and Control Foundation (ISACA), Enterprise Governance is “the set of responsibilities and practices exercised by the board and executive management with the goal of providing strategic direction, ensuring that objectives are achieved, ascertaining those risks are managed appropriately and verifying that the organization’s resources are used responsibly.” The CARS results for each governance related KPA are shown in Figure 18.

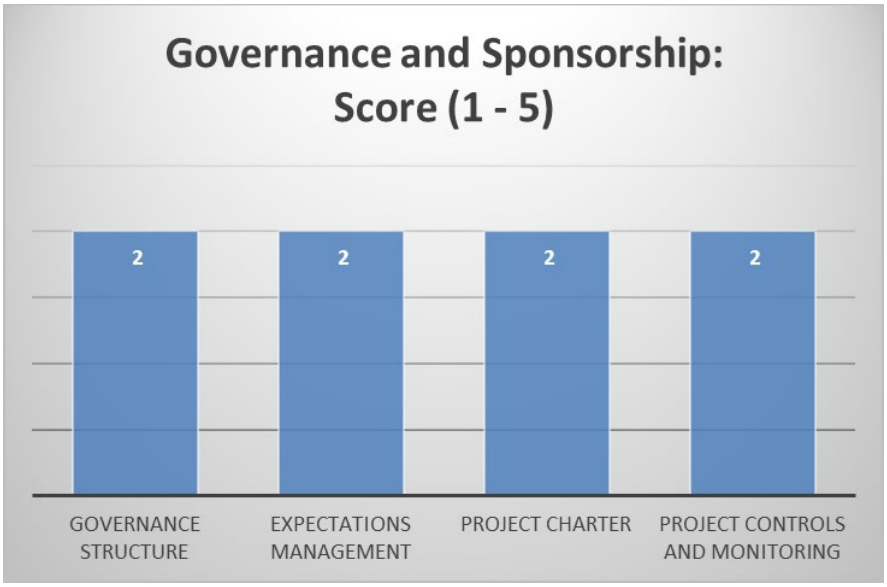

Figure 18: CARS governance and sponsorship scores.

6.3.1 Governance Structure.

Organization or project governance structure describes the arrogation of rights, authorities, accountabilities and responsibilities to key personnel or groups in the organization. A mature governance structure operates according to a standard set of principles, rules, and procedures to increase the effectiveness of decision-making activities. The governance structure addresses code of conduct, strategies, and decision authority across the organization. An organization is more mature when a governance structure is in place and utilized, covering, at a minimum, review, and approval of changes in scope, budget, schedule, and risk.

In the case of the CARS project, the CARS Project Governance Plan is documented and foundationally sound. However, the execution of this governance for an enterprise technology implementation project of this scale was not successful. Foundational components of strategy and architecture for the project were nonexistent. This makes it very difficult to apply governance to a project of this size and complexity.

The CARS project governance plan and related artifacts will be instrumental in the recasting of the CARS project.

6.3.2 Expectations Management.

Expectation Management is a communications discipline typically assigned to Organization Change management or senior leadership which prevents inaccurate assumptions of upcoming change or improvement initiatives. The goal of expectations management is to reduce the risk of initiative failure, stakeholder dissatisfaction and extended change cycles. In some situations, a project can achieve goals and objectives while being perceived as unsuccessful or a failure due to a gap in expectations.

At a minimum, expectation management activities should include an actively engaged Project Sponsor who provides direction and leadership to the project team and to visibility to external stakeholders. Expectation management is only as good as the level of understanding of expectations of all stakeholders.

In the case of the CARS project, expectation management was not a formal practice. Furthermore, the time constraint placed on the project resulted in the project team needing to constrain and/or stifle expectations of stakeholders. Unsuccessful last-ditch efforts of an MVP (minimum viable product) are an example of a shortcoming in this regard.

We did not identify any expectation management related artifacts that are useful going forward.

6.3.3 Project Charter.

The Project Charter is an artifact formulated at or for the official start of a project, initiative, or phase. The approved and published content formally authorizes the existence of the project and provides a referenceable source of the origin of the initiative as well as a description of the desired future outcomes. The Charter provide insight, direction, and a sense of purpose to the management from start to finish.

A project charter names the project manager and defines the authority of the project manager. It gives the project manager the power to utilize organizational resources to accomplish the project objectives. At a minimum, the Project Charter should define the project goals, objectives, scope, exclusions, dependencies, assumptions, constraints, risks (initial), stakeholders, roles, and responsibilities.

In the case of the CARS project, the Project Charter that existed during the project was incomplete and not used as an effective control. Recently, PRD leadership has significantly improved on the project charter and that document will be a cornerstone for recasting the next generation CARS project.

6.3.4 Project Controls and Monitoring.

Project Control and Monitoring processes continually track, review, adjust and report on the project's performance. It's important to measure how a project's performing so that performance can be adequately managed through improvement cycles. These activities provide the greatest likelihood that a successful project stays on track, on budget and on time.

According to the PMBOK® Guide (the Project Management Body of Knowledge), project control is a "project management function that involves comparing actual performance with planned performance and taking appropriate corrective action (or

directing others to take this action) that will yield the desired outcome in the project when significant differences exist."

At a minimum, the project must have a project status dashboard or similar mechanism showing the project status in terms of scope, budget, schedule, and risk. The status should include trends and forecasts that directly represent the effectiveness of project controls.

In the case of the CARS project, project controls from a governance perspective were never successfully implemented. Effective controls would have determined the inferred performance expected was not attainable given the project time and resource constraints. We did not identify any project control related artifacts that are useful going forward.

6.4 Organizational Change Management.

In this section we cover Organizational Change Management (OCM). Organizational change refers to the actions in which an organization or project alters a major component of its structure or operating model, such as its culture, the underlying technologies, infrastructure, regulation, major processes, and/or operating model. OCM must ensure that the organization has the necessary skills, aptitude, and attitude to successfully deploy the new system. Our assessment of CARS in this area is shown in Figure 19.

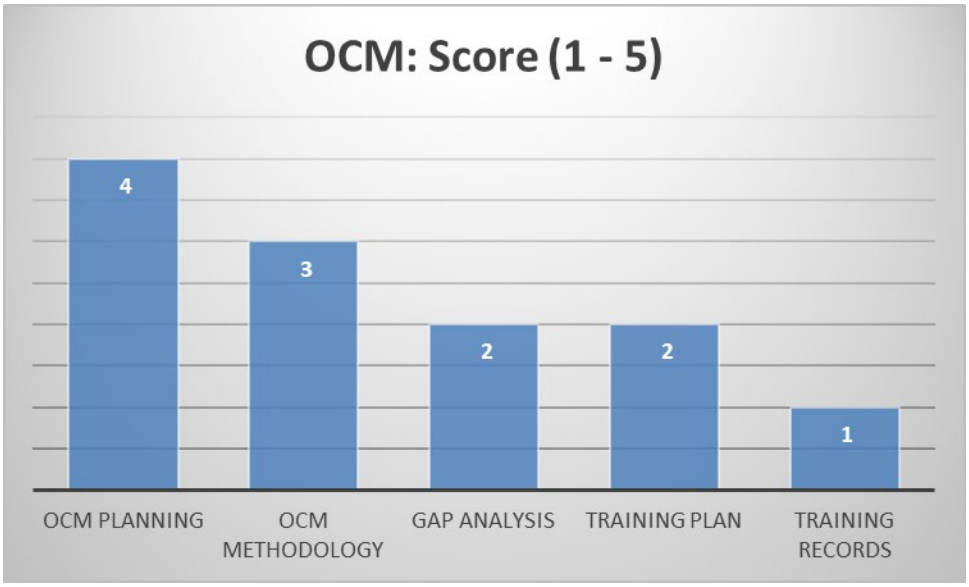

Figure 19: CARS Organizational Change Management Scores

6.4.1 OCM Planning.

OCM planning sets high-level activities traceable to strategic business outcomes, ideally represented in systematic roadmaps in which the strategies appropriately proceed the

changes facing the organization. For success, it is a best practice to ensure OCM plans and strategies address and align key elements of the enterprise ecosystem including stakeholders, resources, timeline, budget, communications, and change.

At a minimum, the organization has an approved OCM plan that identifies and addresses all stakeholders, and that defines the OCM related goals, objectives, activities, resources, and schedule.

In the case of the CARS project, OCM is an established practice with an Organizational Change Management Plan (OCMP) and a variety of related sessions and training. Additional supporting artifacts and activities such as the Change Control Plan defined the CARS Project Change Control activities, processes, and tools.

Also available to the OCM team was The California Department of Technology's "Organizational Change Management Framework". The overview opens with: Organizational Change Management (OCM) is a discipline that helps organizations implement change to achieve measurable results in their business strategies, work processes, structures, technologies, organizational cultures, and management styles. These changes can be driven by implementation of a new mandate, process, technology, or strategy. OCM focuses on the people in the organization and on identifying and enhancing the "human factors" that will achieve and sustain desired project results, which can include improved efficiency, quality, timeliness, and cost avoidance.

We anticipate OCM related artifacts will be useful going forward in the recasting of the CARS project. At minimum, these artifacts and methods should be used to develop the OCM program and plan relative to the new CARS strategy. Early development and continued alignment of the OCM program, plan and activities is essential for project success.

6.4.2 OCM Methodology.

The OCM Methodology is a defined set of activities, artifacts, principles, and techniques intended to streamline enterprise change and increase adoption rates for a given change initiative. The OCM practice space is full of models and methodologies including Bridge, Roger's, Kotter, Kubler-Ross, and Prosci.

In the case of the CARS project, it was not clear which formal methodology, if any, was applied. The project struggled to gain effective definition of success or collaboration across the key stakeholders with respect to OCM, so methods were not applied with any success.

We anticipate the CARS OCM resources, methodology, experience, and associated artifacts will be useful going forward in the recasting of the CARS project. These OCM assets should be engaged early in the CARS strategic planning efforts to ensure alignment.

6.4.3 Gap Analysis.

In the context of OCM, by gap analysis we are referring to an analysis of the skills required to use and support the new system, versus the current skills of the employees. The identified skill gap must then be closed through training. These skill gaps will also be a primary source of resistance to change. The skill gap must cover all stakeholder interactions with the new system, so including for example business users that must use the system, technical staff that must maintain the new system, senior executives that will review outputs from the system, and support personnel that must support the new system.

In the case of the CARS project, the OCM team seemed to primarily focus on the people/resistance side of OCM rather than on the people/skill side of OCM. For example, a common theme from the UAT staff was that they felt thrust into UAT with no understanding of the technical environment or tools. Thus, they were expected to go from a paper-based process to an automated process when they did not have the appropriate technical skills or training. Also, the OCM team seemed to be focused primarily on the business perspective but didn't seem to address the M&O, support, or senior executive perspectives.

We anticipate OCM gap analysis experience and associated artifacts will be useful going forward in the recasting of the CARS project. Specifically, the lessons learned analysis performed by the OCM role we have found consistent with our findings should be utilized during the recasting of the CARS project.

6.4.4 Training Plan.

Once the skill gaps are identified, then a training plan can be created identifying the approach to closing that skill gap. In addition to the obvious things such as who is to be trained, in what, and when; the plan should address approaches to measuring training success and to maintaining and increasing skills over time through refresher or advanced training. In the case of the CARS project, OSaaS seemed to drive the training strategy, plans, curriculums, and training collateral with little or no consideration or collaboration with CARS project OCM team or plans.

We do not anticipate the OSaaS training strategy, planning experience and associated artifacts to be of much use going forward in the recasting of the CARS project.

6.4.5 Training Records.

Once the training plan is defined, records should be maintained to identify who needs to be trained, to track the delivery of that training and the resultant measures of effectiveness, to track requirements for refresher training, and to measure the effectiveness of both the training itself and the trainer.

We did not find CARS related training records, and we did not identify any training records that are useful going forward. We recommend SOS to implement a more robust skills assessment and training program to support its next generation CARS solution.

6.5 Quality Management.

In this section we cover Quality Management. Quality management intentionally engages the organization's stakeholders in activities designed to measure and improve the organization's (or project's) processes, products, services, and culture to achieve the long-term success as determined by defined quality metrics (e.g., stakeholder satisfaction).

Quality management is an industry recognized function across many guidance and standards bodies including ISO/IEEE and PMBOK. Quality management functions to evaluate important aspects of the enterprise project including stakeholder expectations, requirements, and adherence to standards. The quality team establishes quality guidelines, capability maturity targets, business outcomes, metrics, and other factors to determine when and if quality targets and goals are being attained, and related action.

Our assessment of the current (as-is) state with respect to CARS quality management is shown in Figure 20.

Figure 20: CARS quality management scores.

6.5.1 Configuration Control.

Effective quality management requires a clear and consistent understanding of the products and processes to be managed. When documents or code exists in multiple versions, those versions will diverge. When people can change documents or code with no configuration control, at least some of those changes will have unintended consequences, or some people will be working based on the previous version of the

document or code. Simply put, effective quality management is built on the assumption that effective configuration control is in place for all the items that will be managed by the quality processes.

At a minimum, software systems (products) require an approved configuration management plan that is discoverable, accessible, and followed. For more mature organization, the configuration control system should include configuration control processes covering hardware, software, code, engineering artifacts, process documentation, and requirements.

In the case of CARS, the ADO environment itself provides some built-in level of configuration control, but beyond this the project did not seem to formally implement a set of configuration control processes. And in fact, key documents such as requirement specifications did exist in multiple versions and formats. We do not see reusable artifacts in this area.

6.5.2 Process Documentation.

Quality management looks at both processes and products to be sure that performance is to specification. In the case of process quality control, the organizations are measured versus written process documentation. This process documentation is therefore an essential element of quality management. Furthermore, from a quality improvement perspective, process documentation serves to identify opportunities for improvements and remove bottlenecks and inefficiencies. Process documentation is also a key to understanding and training individuals and teams operating in the enterprise.

In the case of the CARS project, process documentation exists but lacked effective management. Artifacts of process documentation were prepared by updating the Perspecta documents but seem to have been ignored or set aside because of deadlines. Another set of process documentation addressing the development and testing work was started but not fully completed. Overall, there are conflicting descriptions of what processes were, or were not, followed, and the corresponding process documentation is not well-organized and ambiguous.

We anticipate process documentation related artifacts will be useful going forward in the recasting of the CARS project. Although many of these process documents and supporting workflow diagrams are dated, these artifacts should be utilized to generate the updated business architecture for the new CARS solution.

6.5.3 Quality Planning.

Quality planning is the process for identifying the quality standards relevant to the project and deciding how and when to achieve quality targets. At a minimum, the organization should have a Quality Management Plan which is available, approved, kept current, and consistently followed.

The Quality Management Plan is essential to a multivendor enterprise technology implementation project. Applying standards-based quality management principles and methods to the program establishes consistent controls and measures across vendors processes, roles, and responsibilities. When applied effectively and pragmatically, this discipline drives objective measurement of performance of the program, project, and resources. We manage what we measure.

In the case of the CARS project, the project created the CARS Quality Management Plan in March of 2018 and refreshed it in the summer of 2020 to support planning, monitoring, controlling, and assuring quality of the CARS project processes and deliverables. The CARS Quality Management Plan is founded on ISO principles and is seemingly sound. However, the plans and related continuous process improvement activities did not appear to be executed effectively for the CARS project. We were unable to discover any evidence to support establishing quality benchmarks or criteria, performing quality assurance activities to ensure conformance to established processes and benchmarks, and performing quality control activities to aid in the measurement of established benchmarks

The CARS Quality Management Plan and related artifacts will be instrumental in the recasting of the CARS project.

6.5.4 Quality Metrics.

A critical step in quality management is adopting quality metrics as standards for measuring the performance of processes and the products they generate. Quality metrics include product metrics, process metrics, and project metrics. Product metrics describe the characteristics of the product such as usability, complexity, design features, performance, and fit for purpose. Process metrics monitor and guide improvement of the development lifecycle, such as defects, defect repair times, response time. Project metrics describe the project characteristics and execution include information such as resource levels, velocity, cost, and schedule.

In the case of the CARS project, the CARS Quality Management Plan provides a solid foundation for definition and management of quality metrics and measurements via the quality management lifecycle. Metrics were to be gathered, analyzed, and discussed each month with key performance indicators rolling into a CARS project dashboard.

Key areas currently measured include the following:

- Issues.
- Risks.
- Schedule.
- Deliverables.
- Training.
- Quality.
- Resources.

- Contract.
- Cost and Budget.

However, the key CARS Metrics and Measurements Strategy identified by the plan was seemingly not developed or not executed effectively for the CARS project.

We anticipate quality metric related artifacts will be useful going forward in the recasting of the CARS project. At minimum, these artifacts provide a strong initial framework for quality management measurement. The key will be to ensure they are implemented, monitored, and acted on.

6.5.5 Quality Control.

Quality control is a routine and continuing effort to maintain the integrity and reliability of enterprise assets (e.g., processes, products), to achieve an expected outcome. Quality control includes oversight of both the organization's processes and products, and products includes both the software being developed and the engineering artifacts to support that development. Product quality control overlaps with the IV&V function, if an IV&V vendor is on board.

In the case of the CARS project, the CARS Quality Management Plan provides a solid foundation for definition and management of quality control. The CARS Quality Management Plan defines Quality Control as the ongoing process of monitoring and recording results of executing the quality activities to assess process performance and recommend necessary changes. Inputs include individual plans or processes, quality metrics and checklists, and process work products. Outputs include quality control defects and measurements, validated changes, identified plan and process updates, and document updates. Quality control is the process of assessing or reviewing process work products and determining whether they comply with the defined criteria.

However, the quality control process did not appear to be executed effectively for the CARS project. We did find any evidence of ongoing process of monitoring and recording results of executing the quality activities to assess process performance and recommend necessary changes.

We anticipate quality control related artifacts will be useful going forward in the recasting of the CARS project. Consistent with the overarching Quality Management theme, at minimum, these artifacts provide an initial framework for quality control. The key will be to ensure the process is implemented, monitored, and acted on.

6.5.6 Configuration Audits.

Configuration Audits are used by the quality assurance team to ensure that there is only one approved, current version of each document and code component, and that the entire project is working from that approved version. Configuration audits are normally conducted as part of the on-going quality control process.

In the case of the CARS project, we did not identify any configuration audit related activity or artifacts that were undertaken, or that are useful going forward.

6.5.7 Process Control.

Process Control is used to effectively manage and optimize on-going processes during the project, as opposed to one-time activities. So, for example, development of the Software Requirement Specification would be a one-time effort, while the software release process might be executed several times during the life of the project (and then on-going during M&O). So, process control techniques would apply to optimizing the release management process, based on lessons learned during each release cycle. Techniques such as DMAIC (Define, Measure, Analyze, Improve, Control) are applied to controlling these recurring processes.

We did not see evidence of process control on CARS, and we did not identify any reusable artifacts in this area.

Appendix A: Acronyms

ADA: Americans with Disabilities Act.

ADO: Azure DevOps.

API: Application Programming Interface.

AWS: Amazon Web Services.

CAL-ACCESS: California Automated Lobbyist and Campaign Contribution and Expenditure Search System.

CalHEERS: California Healthcare Eligibility, Enrollment, and Retention System.

CARES: California Automated Response and Engagement System.

CARS: CAL-ACCESS Replacement System.

CASE: Communications and Stakeholder Engagement.

CCSAS: California Child Support Automation System.

CDSS: California Department of Social Services.

CDTFA: California Department of Tax and Fee Administration.

CHHS: California Health and Human Services.

CMAS: California Multiple Award Schedule.

CMMI: Capability Maturity Model Integrated.

CMS: Case Management System.

COTS: Commercial Off-The-Shelf.

CPI: Cost Performance Indicator.

CPU: Central Processing Unit.

CROS: Centralized Revenue Opportunity System.

CWS: Child Welfare System.

DBMS: Database Management System.

DDR: Detailed Design Review.

DMAIC: Define, Measure, Analyze, Improve, Control.

EAC: Estimate at Completion.

EDR: Enterprise Data to Revenue.

eMRI: enterprise Maturity Readiness Index.

ETC: Estimate to Complete.

ETL: Extract-Transform-Load.

FPE: Function Point Equivalent.

FPPC: Political Practices Commission.

FTB: Franchise Tax Board.

GAAP: Generally Accepted Accounting Procedures.

HLO: High-Level Object.

ID: Identifier.

IEEE: Institute of Electronic and Electrical and Electronics Engineers.

IPS: Integrated Project Schedule.

ISACA: Information Systems Audit and Control Foundation.

ISO: International Standards Organization.

IT: Information Technology.

ITD: Information Technology Division.

IV&V: Independent Verification and Validation.

KPA: Key Process Area.

LEADER: Los Angeles Eligibility, Automated Determination, Evaluation and Reporting.

M&O: Maintenance and Operations.

MOU: Memorandum of Understanding.

NIST: National Institute of Standards and Technology.

OCM: Organizational Change Management.

OCMP: Organizational Change Management Plan.

ODC: Other Direct Charge.

OS: Operating System.

OSaaS: Outreach Solutions as a Service.

PAL: Project Approval Lifecycle.

PERT: Program Evaluation and Review Technique.

PIER: Post Implementation Evaluation Review.

PMBOK: Project Management Body of Knowledge.

PMO: Project Management Office.

PRA: Political Reform Act.

PRD: Political Reform Division.

PRR: Production Readiness Review.

R&D: Research and Development.

RFO: Request for Offer.

RMP: Risk Management Plan.

ROM: Rough Order of Magnitude

SAWS: Statewide Automated Welfare System.

SCSEM: Safeguard Computer Security Evaluation Matrix.

SDLC: Software Development Lifecycle.

SIT: System Integration Testing.

SME: Subject Matter Expert.

SOS: Secretary of State.

SPI: Schedule Performance Indicator.

SRR: Software Requirement Review.

STP: Software Test Plan.

T&M: Time and Material.

TID: Tax ID.

TRR: Test Readiness Review.

UAT: User Acceptance Testing.

UI: User Interface.

Appendix B: Assumptions

The ExcelerPlan benchmark driven modeling tool is configured using project specific environmental variables that have an impact on both efficiency and non-linear impacts of economies and diseconomies of scale. These specific variable settings are being used to develop the CARS forecasts of cost, schedule, and labor requirements that will be delivered as part of the CARS Roadmap Forward (WOA #4). The variables then form the assumptions that underly that plan. Some of these variables (e.g., Precedence) are simply project characteristics. Others (e.g., Language and Tool Experience) are variables that can and should be confirmed through the acquisition process via the structure of the RFP requirements. The variable settings that we propose to use are shown in Table 7, along with the justification for each setting. The values (Very Low, Low, Nominal, High, or Very High) for each setting are used as one of the inputs by the estimation models to develop the ROM budgets, schedule, labor requirements, and so on.

The ROM estimation modeling overall architecture is shown in Figure 21. The project scope is defined using High Level Objects, as discussed earlier, and illustrated with the example in Table 6. Project specific environment variables are then used as adjusting put to the project estimation, and it's those variable settings that we are reviewing in Table 7. With those inputs defined, the ExcelerPlan models will then compute a system size in Function Point Equivalents (FPE), then use historic project benchmark data, expressed as parametric curves, to compute the total forecast project cost, schedule, and level of effort. Historic allocation templates for other similar projects in California are then used to allocate the effort and cost to the various groups that will be working on the project.

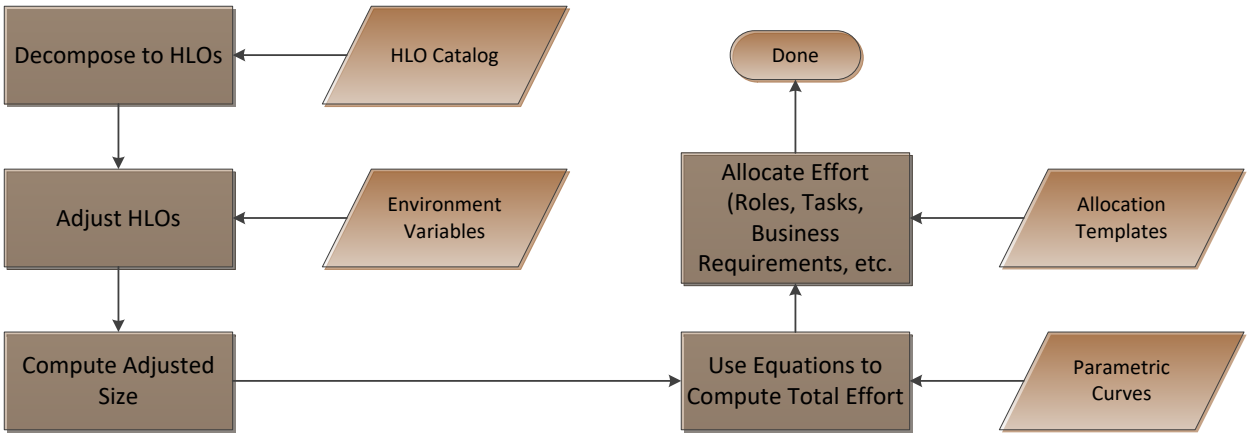

Figure 21: ROM estimation modelling approach.

Note that these settings are *not* an evaluation of the existing CARS project with respect to these variables. Rather, they are the expected situation for the CARS Restart going forward, and thus the basis of our estimates for that future work. These settings were reviewed with the SOS CARS Project Sponsor to align expectations.

Table 7: Assumptions for CARS Roadmap Forward estimate.

Variable	Description	CARS Setting
Pace	Pace of development relative to normal project in similar organizations (VH is the maximum theoretical pace without reducing scope).	Nominal. Pace was set to match previous successful projects in the benchmark data.
Overlap	Degree of task overlap planned for in the baseline schedule. N=None; H = 10%; VH = 25%.	High. This setting was selected because it balances the need for CARS to be operational as quickly as possible with the risk and cost of accelerated development.
Plans & Requirements	Time spent on up-front planning and requirements relative to a benchmark project.	Very Low. CARS has already done a lot of the up-front requirement related work to quite a good level of detail. This is an area where the CARS restart will be saving money and schedule relative to the initial CARS effort.
Integration and Test	Time spent on integration and test (post unit test) activities relative to a benchmark project.	Nominal. The CARS integration and test effort should be typical of similar efforts.
Req. Volatility	Degree to which requirement changes/clarifications are absorbed by the project budget. Nominal is 7%.	Nominal. The CARS project is expected to be typical in terms of requirement volatility during development.
Team Cohesion	Degree to which stakeholders are cooperative. Nominal is basically cooperative. VL means very difficult interactions.	Low. There is the opportunity for this to move to either Nominal or High, based on the degree to which all stakeholders align with the project vision and are fully supportive of the project and each other.
Requirements Flexibility	Degree to which requirements can be relaxed based on technical challenges. Nominal is some relaxation. VL means rigorous requirements.	Nominal. We believe that the project has a typical mix of rigid and flexible requirements. Note that flexibility in this context does not mean that legislative requirements can be relaxed.
Project Risk	Degree to which project problems result in business risk, nominal is typical, very high is high business risk.	High. This setting is based on the risk associated with CalACCESS potentially failing prior to the project being prepared to replace that system.
Precedence	Degree to which the project may be considered R&D. VH is largely familiar, VL has many R&D characteristics.	Very High. This is another area where the work that CARS has already done has clarified the requirements and the required approach, therefore resulting in reduced costs and schedule going forward. To a large extent, CARS has performed exploratory Research and Development (R&D) to determine if a Commercial Off-The-Shelf (COTS) transfer product would meet the California needs (no), and to determine if Salesforce would meet the California needs (also no).
Process Maturity	Capability Maturity Model Integrated (CMMI) Level: VL=Level 1 lower half; L = Level 1 upper half; N = Level 2; H = Level 3; VH = Level 4 or 5	This variable measures the degree to which effective processes are in place and followed by the project. The OSaaS implementation setting would be Low, however going forward we believe that the SOS will be implementing most of the eMRI recommendations that we have included in our report. Based on

Variable	Description	CARS Setting
		that assumption, we are using a setting of High. Again, this is an area where the lessons-learned from the first two CARS iterations will now result in a cost and schedule saving going forward.
Availability of SMEs	Degree to which technical and business experts are available. VH means readily available.	We believe that the correct setting is High, which balances the need of the SMEs to continue with their on-going work versus the requirements for support to the project. This is another area where we believe that the CARS restart will benefit from the project experience thus far, resulting in a cost and schedule saving going forward.
Applications Experience	Average team experience with the application. N=1 year; VL= 2 months; L=6 months; H= 3 years; VH=6 years.	We believe that the correct setting is Nominal. This setting is based on the balance of SME experience, experience by external stakeholders (e.g., Netfiler), and the experience of the project implementation team. We believe that the OSaaS effort was Very Low, in that the project had a very large team of people with zero application domain experience and a relatively low offsetting pool of people with the necessary application experience. Again, this is an area where the lessons learned from the previous iterations will result in cost savings going forward. We recommend that the amount of application (California specific filing) experience by team members of the SI vendor be included as a scored desired qualification. Experience with taxation systems may be relevant as well and should be counted. We do not believe that it should be a mandatory requirement, however.
Language and Tool Experience	Average experience with the programming language/tools. N=1 year; L=6 months; VL=2 months; H=3 years; VH=6 years.	This setting applies specifically to the programmers. We recommend that developers working on this have a minimum of 3 years of previous experience working with the programming language/tools they will be working with, resulting in a setting of High.
Platform Experience	Average virtual machine experience (Operating System (OS), DBMS, etc.). N=1 year; L=6 months; VL=2 months; H=3 years; VH=6 years	This setting applies to the experience of the technical team (programmers and database developers) with the DBMS, operating system, and other virtual machine stack components. For example, if the solution was built using Amazon Web Services (AWS), it would include experience working with AWS. We recommend requiring that the technical team have a minimum of 1 year of experience working with the virtual machine proposed for implementation, resulting in a setting of Nominal.
Static/Dynamic Ratio	Data Conversion Only: VL=15%; L=35%; N=55%; H=75%; VH=90%	This variable applies to the data conversion. We believe that 90% or more of the CAL-ACCESS data to be converted is static, resulting in a setting of Very High.

Variable	Description	CARS Setting
Acquisition Type	N=New or Replace; L=Upgrade	Nominal.
Acquisition Stakeholders	N=Few; H=Diverse (e.g., multiple States, multiple counties)	Nominal.
Acquisition Funding	L=Benefit; N=State; H=Federal + State	Nominal.
Procurement Approach	N=One-Step; H=Two Step. One-step acquisitions are traditional acquisitions, while two-step procurements award multiple vendors a small proof-of-concept or discovery contract; followed by a negotiated contract with one of the vendors from that initial acquisition.	Nominal.
Acquisition Speed	N=Typical; L=Fast, efficient processes; H=Slow, inefficient processes.	Low: We believe that the acquisition will be fast-tracked.
Tier Level	VL=Tier 5 or 4; L=Tier 3; N=Tier 2; H=Tier 1; VH=Mainframe/Core	Nominal: We believe that the server environment is properly classified as a Tier 2 application from a Recovery Time Objective perspective.
Hardware Loading	User interaction loading, VL to VH	Nominal: The user interaction loading characteristics are typical.
Database Loading	Loading on database server, VL to VH	Nominal: The database loading characteristics are typical.
Infrastructure Integration	Computer-to-computer transaction loading, VL to VH	Nominal: The interface and batch processing characteristics are typical.

Appendix C: eMRI Evaluation Framework

Table 8 contains the eMRI model used to evaluate the CARS project. This model is used to evaluate the performance of the entire CARS project team in the execution of the project. While the primary organization evaluated is the assigned system integrator (OSaaS, in this case), the evaluation also includes other stakeholder groups within the organization that play key roles with respect to system integration activities (e.g., governance activities performed by the State stakeholders). The model is broken down into categories and key process areas, or KPAs. For each KPA, the demonstrated project capabilities will be scored somewhere between Very Low (area of weakness, no repeatable process strength) and Very High (area of strength, repeatable and optimum processes). The specific criteria used when assigning this score are also shown in the table.

Table 8: eMRI project evaluation model.

Category	KPA	Very Low	Low	Average	High	Very High
Data Conversion and Migration	**Data Integrity**	There are no records available confirming that the migrated data matches the source data.	Spot checking is used to verify data integrity.	Summary comparisons of the source and the target data are available, supplemented by spot checking at the record level.	Summary and detail data validation records are available, but those records are not sufficient to meet audit criteria.	Auditable records exist demonstrating that 100% of the source data was migrated, with no data loss or unplanned data changes.
Data Conversion and Migration	**Data Quality**	The migrated data is lower in quality than the source data in that records were changed during the process and in some cases those changes introduced errors.	Most of the data has been migrated, and the quality of the migrated data is the same as, or like, the quality of the source data.	All data has been migrated, and the quality of the migrated data is the same as, or like, the quality of the source data.	All data has been migrated, with some data clean-up during the process, but some unresolved problems remain.	All data is migrated, cleansed and usable with duplicate data resolved and removed.
Data Conversion and Migration	**Data Control**	Business rules that automatically change data are executed ad hoc	Business rules that automatically change data are documented,	Some business rules that automatically change data were	All business rules that automatically change data were validated by the data owner(s).	All business rules that automatically change data were validated by the data owner(s)

Category	KPA	Very Low	Low	Average	High	Very High
		by the technical team with little or no documentation.	but not validated by the data owner(s).	validated by the data owner(s).		and documentation of that approval exists.
Data Conversion and Migration	**Data Security**	No formal approach to security of the data to be migrated is documented or enforced.	Data security involves physical access control only.	Data at rest is under controls equivalent to those used in the production system	Data at rest and in motion is under controls equivalent to those used in the production system	Data at rest and in motion is under controls equivalent to those used in the production system, including logging and monitoring.
Data Conversion and Migration	**Conversion Planning**	Data conversion planning is ad hoc and informal only.	Data conversion planning is integrated with the other project management documentation, and incomplete.	Data conversion work is defined, but the plan elements are included in multiple documents/ locations.	A data conversion plan exists covering conversion strategy, assumptions, constraints, activities, including activities, resources, roles, responsibilities, timeline, milestones, and risks. The plan is not formally approved or is not maintained.	An approved data conversion plan exists covering conversion strategy, assumptions, constraints, activities, including activities, resources, roles, responsibilities, timeline, milestones, and risks. The plan is maintained and current.
Data Conversion and Migration	**Conversion Specifications**	Conversion related requirements are ad hoc or developed "just in time."	Data conversion specifications exist, but they are incomplete.	Data conversion specifications exist, but they are neither approved nor maintained.	Data conversion specifications exist, but they are not approved, or they are not maintained.	Approved data conversion specifications document target and source data tables with mapping, data volume (e.g., record counts), field mappings, transform rules, validation requirements, and differentiating static

Category	KPA	Very Low	Low	Average	High	Very High
						from dynamic source data. The document is maintained and current.
Data Conversion and Migration	**Conversion Architecture**	Data is converted using ad hoc resources and tools with little or no documentation.	Data is converted using ad hoc but documented resources and tools.	Data is converted using shared resources and dedicated tools, with incomplete architectural documentation.	Data is converted using dedicated resources and tools, but there are questions about the adequacy of the resources to meet the conversion requirements.	A data conversion architecture exists and provides adequate resources to meet the conversion requirements.
Data Conversion and Migration	**Conversion Scheduling**	Conversion is scheduled to be continuing during UAT.	An initial full data conversion is scheduled for completion prior to the start of UAT. Final cutover conversion will require 75% or less of the available cutover conversion window.	An initial full data conversion is scheduled for completion prior to the start of SIT and performance testing. Final cutover conversion will require 50% or less of the available cutover conversion window.	Conversion of static data is scheduled prior to Unit Testing, and an initial full data conversion is scheduled for completion prior to the start of SIT and performance testing. Final cutover conversion will require 50% or less of the available cutover conversion window.	Conversion of static data is scheduled prior to Unit Testing, and an initial full data conversion is scheduled for completion prior to the start of SIT and performance testing.

Category	KPA	Very Low	Low	Average	High	Very High
Data Conversion and Migration	**Conversion Cutover**	Cutover success and failure criteria are not defined, and there is no approved rollback strategy in place.	The final cutover conversion time relative to the available window is not known, but cutover success and failure criteria are defined, and an approved rollback strategy is in place.	Final cutover conversion will require 75% or less of the available cutover conversion window, cutover success and failure criteria are defined, and an approved rollback strategy is in place.	Final cutover conversion will require 50% or less of the available cutover conversion window, cutover success and failure criteria are defined, and an approved rollback strategy is in place.	Final cutover conversion will require 25% or less of the available cutover conversion window, cutover success and failure criteria are defined, and an approved rollback strategy is in place.
Contract Management and Vendor Negotiations	**Training**	Government staff responsible for contract management have inadequate contract training, and available resources for self-learning are inadequate or unavailable.	Government staff responsible for contract management are self-taught "just-in-time", using resources that are available.	Government contract staff responsible for contract management are self-taught, with training monitored but not required.	Both government contract and technical staff have received suitable contract management training, but there are no formal requirements for refresher training.	Government contract and technical staff have received suitable contract management training, including annual refresher training.
Contract Management and Vendor Negotiations	**References**	Contractor references are not checked prior to contract award.	Contractor references are spot-checked prior to contract award, and informally incorporated into the evaluation criteria.	Contractor references are checked prior to contract award. Results are incorporated as part of the other evaluation criteria.	Contractor references are checked prior to contract award. Results are a scored criteria in the evaluation.	Contractor references are checked, including secondary references, prior to contract award. Results are a scored criteria in the evaluation.
Contract Management and Vendor Negotiations	**Vendor Evaluations**	There is no formal or informal vendor evaluation process.	Vendor performance is occasionally evaluated.	Informal vendor evaluations are provided for most contractors.	Written contractor evaluations are prepared for contractors at the conclusion of all contracts.	Written contractor evaluations are prepared for contractors at the conclusion of all contracts, and

Category	KPA	Very Low	Low	Average	High	Very High
						annually for contracts extending beyond one-year.
Contract Management and Vendor Negotiations	**Contract Negotiations**	Contract negotiation is not part of the procurement process.	Contracts are occasionally negotiated if there is an obvious opportunity for cost saving.	Contracts are regularly negotiated to offer the best value to the government, but government negotiators have no negotiation training.	Contracts are regularly negotiated to offer the best value to the government. Government negotiators have negotiation training.	Contracts are regularly negotiated to offer the best value to the government. Government negotiators have negotiation training. Independent government cost estimates are prepared to support vendor fair and reasonable analysis and negotiation.
Contract Management and Vendor Negotiations	**Contract Change Management**	Vendors perform significant work on contracts without formal written authorization by the government contracting officer.	Vendors perform significant work on contracts prior to formal written authorization by the government contracting officer.	Vendors perform small amounts of work on contracts based on verbal authorizations, with formal written authorization in place prior to invoicing.	All contractual changes are approved by the government contracting officer in writing, with no expenditures prior to written approval.	All contractual changes are approved by the government contracting officer and the Project Sponsor in writing, with no expenditures prior to written approval.
Contract Management and Vendor Negotiations	**Competition**	Contracts are awarded sole source with no formal justification or approval of the sole source decision. Or contractors are awarded	Contracts are frequently awarded sole source with justification and approval of that decision.	Contracts are generally awarded competitively, but with limited competition (less than 3 bidders) a common occurrence.	Contracts are awarded and administered in compliance with regulatory guidelines, full and open competition, and ethical best practices.	Contracts are awarded and administered in compliance with regulatory guidelines, full and open competition, and ethical best practices. Independent

Category	KPA	Very Low	Low	Average	High	Very High
		competitively, but the structure of the procurement prevented adequate competition.				oversight is in place and effective to ensure compliance of the contracting process.
Contract Management and Vendor Negotiations	**Contract Risk Allocation**	The government assumes significant contract risk for areas where it has little or no control.	Risk is allocated to both the government and contractor, with no formal approach to risk allocation between the contracting parties is in place.	Most risks area allocated to the contractor, even in situations where the contractor has little or no control over the risk factors.	Risks are mostly structured to allocate the risks to the party best able to control the relevant risk factors.	Contracts are consistently structured to consistently allocate the risks to the party best able to control the relevant risk factors.
Contract Management and Vendor Negotiations	**Deliverable Management**	Contractual deliverable specifications are incomplete or vague, or deliverables are modified significantly on the fly during project execution. Deliverables are accepted by individuals other than the government contracting officer and/or designated government technical representative.	Contract SOWs clearly define all deliverables due under the contract. These deliverables are comprehensive and verifiable. Deliverable approval is ad hoc and informal.	Contract SOWs clearly define all deliverables due under the contract. These deliverables are comprehensive and verifiable. Deliverable approval is mostly in writing but sometimes informal.	Contract SOWs clearly define all deliverables due under the contract. These deliverables are comprehensive and verifiable. Contract deliverables are approved in writing by one government employee, one for technical and one for contractual compliance. Procedures are in place and used to ensure that all contractual deliverables are	Contract SOWs clearly define all deliverables due under the contract. These deliverables are comprehensive and verifiable. Contract deliverables are approved by two government employees, one for technical and one for contractual compliance. Procedures are in place and used to ensure that all contractual deliverables are received and approved.

Category	KPA	Very Low	Low	Average	High	Very High
					received and approved.	
Requirement Definition and Management	**Functional Requirements**	Less than 50% of the detailed functional requirements exist, and there are conflicting views about the high-level functional requirements.	Less than 50% of the detailed functional requirements exist, but there is consensus about high level functional requirements.	75% of the detailed functional requirements exist.	95% of the detailed functional requirements exist.	An approved and comprehensive set of testable functional requirements exist that fully describe the required scope of work.
Requirement Definition and Management	**Non-Functional Requirements**	Less than 50% of the detailed non-functional requirements exist, and there are conflicting views about the high-level non-functional requirements.	Less than 50% of the detailed non-functional requirements exist, but there is consensus about high level non-functional requirements.	75% of the detailed non-functional requirements exist.	95% of the detailed non-functional requirements exist.	An approved and comprehensive set of testable non-functional requirements exist that fully describe the required scope of work.
Requirement Definition and Management	**Requirement Structure**	Requirements are unstructured with no defined nomenclature or categorization.	Requirements are categorized, but the categories do not align with the business functions.	Requirements are organized into categories and those categories	Requirements are organized into categories and those categories align with business functions.	Requirements are organized into a hierarchical structure that aligns with the business functions.

Category	KPA	Very Low	Low	Average	High	Very High
				somewhat align with business functions.		
Requirement Definition and Management	**Requirement Storage**	Requirements exist in a variety of places and formats. They are sometimes inconsistent. Not all requirements are in a readily accessible format.	Requirements are in a variety of places, but they are all electronically accessible and a requirement "system of record" defines which requirements are official.	Requirements are assigned a unique ID and stored electronically.	Requirements are assigned a unique ID, stored electronically, and managed with configuration or version control.	Requirements are assigned a unique ID, stored electronically, and managed with both configuration and version control.
Requirement Definition and Management	**Traceability**	Relationships between different requirement sources and level is haphazard.	Requirements are traced through a categorization or clustering approach.	Detailed requirements are traceable backward to their source.	Requirements are traceable forward and backward, but traceability is not always verified and approved.	Requirements are traceable both backward to their source and forward to derived requirements. Reports are available and used to ensure full requirement coverage. Traceability is verified and approved.
Requirement Definition and Management	**Requirement Status**	Project status tracking is largely independent of requirement tracking and management.	The status of high-level requirements is generally tracked.	High level requirements are tracked through SIT and UAT, with test case status substituting for requirement tracking during testing.	Detailed requirements are tracked through the SIT and UAT activities.	The status of each requirement is tracked through design, implementation, unit testing, SIT, UAT, and final approval.

Category	KPA	Very Low	Low	Average	High	Very High
Requirement Definition and Management	**Requirement Management Processes**	Requirement management is ineffective.	Requirement management is largely effective but driven by the effort of individuals rather than processes.	Some requirement management processes are defined, others are ad hoc. Processes are generally followed, but with many exceptions.	Processes are managed, but with specific non-critical areas of weaknesses.	Processes for managing requirements and controlling changes to requirements exist, are documented, and are consistently followed.
Requirement Definition and Management	**Business Process Improvement**	Documented requirements do not adequately represent critical existing business processes.	Documented requirements capture most existing business processes, but some less critical processes are either missed or very high level.	Documented requirements capture existing business processes accurately.	Documented requirements capture existing business processes and define improvements in some of the most critical areas.	The requirement process includes a specific process improvement step during which expected improvements because of the project underway are identified and defined.
Project Schedule Management	**Integrated Project Scheduling**	Project schedule(s) are significantly incomplete or inaccurate.	Different groups use varying degrees of formal project scheduling, and there is minimal attempt to integrate those schedules.	A milestone level integrated project schedule exists, but detailed scheduling is the responsibility of individual groups within the project.	An integrated project schedule exists and defines work for most project groups, but the work of some groups is managed outside of that schedule.	An integrated project schedule exists, captures all the required project work, assigns work to individuals or groups, and includes task dependencies.
Project Schedule Management	**Milestone Reviews**	The project has no scheduled milestone reviews. Project schedules may show many milestones, but those are delivery or capability dates	The project has milestone reviews for the most critical milestones, at the end of the project (e.g., Production Readiness Review), but no formal reviews during	The project has milestone reviews for the major stages, but also has extended periods of time with no milestone reviews scheduled.	Milestone reviews are defined and occur at most every six-months or 25% of the project duration, whichever is less.	Milestone reviews are defined and occur at most every six-months or 25% of the project duration, whichever is less. These are gate reviews, requiring

Category	KPA	Very Low	Low	Average	High	Very High
		rather than formal reviews.	the development process.			formal approval to proceed.
Project Schedule Management	**Schedule Updates**	The schedule is not current/ accurate.	The schedule is updated on an as-needed basis.	The schedule is updated on a regular basis, and those updates are distributed.	The schedule is updated on a regular basis, and updates are approved through a governance process.	The schedule is updated on a regular basis, and updates are approved through a governance process. A baseline schedule exists and trends relative to that baseline are maintained and published.
Project Schedule Management	**Project Status**	Actual and status information is not accurately captured.	Actual and status information is partially captured.	Actuals and status information is captured in the schedule, and that information is then distributed (e.g., in a Gantt chart with status).	Actuals and status information is captured in the schedule and used to compute and report a project dashboard.	Actuals and status information is captured in the schedule and used to compute, and report earned value data including SPI, CPI, Estimate to Complete (ETC), and Estimate at Completion (EAC).
Project Schedule Management	**Estimating**	The project timeline and resource requirements are incorrect.	The project timeline and resource requirements are partially accurate and partially inaccurate or low confidence.	Project timeline and resource requirements used bottom-up estimates by the individuals/ groups responsible for the work.	Project timeline and resource requirements used bottom-up estimates using the PERT or Delphi technique.	The project timeline and resource requirements used benchmark data to validate the plan.

Category	KPA	Very Low	Low	Average	High	Very High
Project Schedule Management	**Resourcing**	Resource requirements are not clearly known or are not tied to the schedule.	Many resources are overloaded to > 150%.	Some resources are overloaded to >150%	Some resources are overloaded to > 125%.	Resource leveling was used to adjust the schedule to actual resource availability and constraints.
Communication Management	**Communications Strategy**	Communications are ad-hoc or significantly limited to both internal and external stakeholders.	Communications strategies include internal communications based on various meetings and reports with limited stakeholder group distinction to communications	Communication Strategy is documented and includes processes, formats, and reporting specification for all stakeholders.	Communication strategy is documented and managed by internal staff who maintain communications frequency and completeness.	Approved Communication strategy includes specific communication specification for each stakeholder group, including Signed MOUs for interface partners and other external stakeholders with direct project responsibilities.
Communication Management	**Communication Planning**	Communications are ad-hoc and reactive with little or no planning or strategic value.	Communications plan is initiated and covers 25% of the stakeholder community, with semi standard communications	Communication Plan is completed and covers all stakeholders with the definition of standard communications	Communication plan is complete and in use to guide all stakeholder communications.	A Communication Plan exists, is approved, is maintained, and followed for all defined stakeholders.
Communication Management	**Stakeholder Definition**	Stakeholders are not fully identified	Stakeholder definition is standardized, and Stakeholders are being identified.	The project team manages a single stakeholder list which includes identified individuals.	A comprehensive stakeholders list exists but is not fully populated.	A comprehensive stakeholder list identified individuals and roles with contact information exists, is complete, and is maintained.

Category	KPA	Very Low	Low	Average	High	Very High
Communication Management	**Project Status Communication**	Project status is verbal, ad-hoc or discussed in status meetings.	Project status communication are schedule, standardized and delivered to management.	Project status communications are scheduled, standardized, and delivered digitally. Prior status reports are available on request.	Project status communication is published digitally and can be accessed through a portal or received through email and delivered at regularly scheduled status meetings.	A project website, webpage, or other similar mechanism exists and contains up to date project status. Written project status reports are submitted at least month, are accurate, and are comprehensive. Project status review meetings are held regularly with project leadership, business, and project team.
Communication Management	**Action Item Management**	Action Items are managed ad-hoc with little or no follow up reporting	Action Items are submitted or discovered at regular status meetings and assigned to team members	Action Items are logged and categorized to improve assignments. Action Item close out is reported.	A Project Action item register is used to record current and completed action items.	A project action item register is available, used, and current. Action items are assigned with due dates and status.
Communication Management	**Meeting Management**	Meetings are on an ad-hoc basis with little or no formal structure.	Standard meetings are owned and scheduled with consistency.	Meetings are scheduled, owned, and follow a specific format including agendas, purpose, and minutes	Agendas are prepared for meetings, and minutes for meetings are prepared and distributed after each meeting. Action items, issues, and risks that are identified during meetings are documented both in the meetings and in	In addition to the attributes of High score, Meeting information is managed centrally with role-based accessibility to past current and future meetings including Agendas, minutes, action items, issues, and risks.

Category	KPA	Very Low	Low	Average	High	Very High
					the action item and risk registers.	
Governance and Sponsorship	**Governance Structure**	Little or no formal governance structure exists across the project.	A defined formal structure exists but is not yet adopted or followed.	A governance structure exists and is used for all decisions.	A governance structure is in place and utilized, covering as a minimum review and approval of changes in scope, budget, schedule, and risk.	A governance structure is in place and utilized, covering as a project level (review and approval of changes in scope, budget, schedule, and risk), business level (review and approval of process, performance, or policy changes), and technical level (review and approval of technology, infrastructure, or data changes).
Governance and Sponsorship	**Expectations Management**	Expectations are assumed or understood through the project requirements.	External stakeholders are surveyed to discover expectations that need to be considered in project scope.	Expectations are research and recorded, reviewed, and approved as in scope, and traceable to project requirements.	Expectations are actively reviewed and used to guide project activities, designs, and solutions. The Project Sponsor is actively engaged in providing communications to the project and to	The Project Sponsor is actively engaged in providing leadership to the project members and external stakeholders. Expectations are frequently reviewed, amended or achieved by improving the solution or products.

Category	KPA	Very Low	Low	Average	High	Very High
					visibility to external stakeholders.	
Governance and Sponsorship	**Project Charter**	The project does not have a charter to guide governance or direction of the project.	Project charter is under development to incorporate project purpose goals and scope.	Project charter exists and defines the project purpose, goals, and scope.	Project Charter exists and defines the project goals, objectives, scope, stakeholders, roles, and responsibilities.	An approved Project Charter exists and defines the project goals, objectives, scope, exclusions, dependencies, assumptions, constraints, risks (initial), stakeholders, roles, and responsibilities.
Governance and Sponsorship	**Project Controls and Monitoring**	Limited or no formal project controls exist. Project monitoring is on a simple task completion report.	Project controls are established for scope, budget, schedule, and risk.	Project controls are established for scope, budget, schedule, and risk with measurements taken to monitor project alignment with intended outcome.	A project status is managed and reported including scope, budget, schedule, and risk. The status includes trends and forecasts. As well as a descriptive explanation of the controls in use.	A project status dashboard or similar mechanism exists showing the project status in terms of scope, budget, schedule, and risk. The status includes trends and forecasts.
Organizational Change Management	**OCM Planning**	OCM is identified as a requirement, but no formal planning related to OCM has been conducted.	Some OCM related planning documents exist, but they are scattered and informal (e.g., in PowerPoint presentations).	An OCM plan exists but it is not comprehensive and not fully maintained.	An approved OCM plan exists, addresses all stakeholders, and defines as a minimum the OCM related goals, objectives,	An approved OCM plan exists, addresses all stakeholders, and defines as a minimum the OCM related goals, objectives, activities, resources,

Category	KPA	Very Low	Low	Average	High	Very High
					activities, resources, and schedule.	and schedule. The plan is maintained and current.
Organizational Change Management	**OCM Methodology**	No standard OCM methodology is planned, and OCM staff have limited OCM experience.	No standard OCM methodology is planned, but OCM staff have previous OCM experience.	A standard OCM framework, such as ADKAR, is planned. OCM staff have limited experience with the methodology, but support materials are available.	A standard OCM framework, such as ADKAR, is followed, and OCM staff have familiarity with that methodology.	A standard OCM framework, such as ADKAR, is followed, and OCM staff are trained and certified in that methodology.
Organizational Change Management	**Gap Analysis**	No gap analysis was conducted.	An informal gap analysis was conducted at a high level and covering most stakeholders.	A gap analysis has been conducted for all business stakeholders, identifying as-is and to-be skills needed both during system development and following deployment.	A gap analysis has been conducted for all internal business and technical stakeholders, identifying as-is and to-be skills needed both during system development and following deployment.	A gap analysis has been conducted for all internal and external business and technical stakeholders, identifying as-is and to-be skills needed both during system development and following deployment.
Organizational Change Management	**Training Plan**	No training plan was prepared to support the OCM effort.	A training plan, or related training documentation, is available but it is not comprehensive and not tied back to the gap analysis.	An approved training plan exists identifying the approach to be followed to close the skill gaps identified in the gap analysis. The training plan includes internal	An approved training plan exists identifying the approach to be followed to close the skill gaps identified in the gap analysis. The training plan includes internal business and technical stakeholders.	An approved training plan exists identifying the approach to be followed to close the skill gaps identified in the gap analysis. The training plan includes internal and external business and

Category	KPA	Very Low	Low	Average	High	Very High
				business stakeholders.		technical stakeholders.
Organizational Change Management	**Training Records**	Training is conducted informally, with few or no training related records.	Training records exist, but they are inconsistent or not complete.	Training records exist for all OCM related training, including attendees and instructor. Certificates of attendance are issued to attendees.	Training records exist for all OCM related training, including attendees and instructor, attendee testing results, and instructor evaluation results. Certificates of attendance are issued to attendees.	Training records exist for all OCM related training, including attendees and instructor, attendee testing results, and instructor evaluation results. Follow-on retention testing is conducted 3 to 6 months following training. Certificates of attendance are issued to attendees.
Quality Management	**Configuration Control**	Project material exists in various locations and versions, with no clear identification of the latest version.	Some combination of Hardware, software, code, engineering artifacts, process documentation, and requirements are not under formal configuration control, but the latest version is clearly identified and recognized by all project participants.	Hardware, software, code, engineering artifacts, process documentation, and requirements are under configuration control.	An approved configuration management plan exists and is followed. Hardware, software, engineering artifacts, process documentation, and requirements are under both configuration control.	An approved configuration management plan exists and is followed. Hardware, software, engineering artifacts, process documentation, and requirements are under both configuration and version control.
Quality Management	**Process Documentation**	Process documentation exists, but it is inaccurate.	Work is performed ad hoc without underlying process documentation, but individuals do have	Some of the most significant processes are documented, and those processes	Project processes and supporting work products (e.g., checklists) covering most areas of work	Project processes and supporting work products (e.g., checklists) covering all areas of work are

Category	KPA	Very Low	Low	Average	High	Very High
			standard approaches to performing their work.	are consistently followed.	are available, approved, understood, and consistently used.	available, approved, understood, and consistently used.
Quality Management	**Quality Planning**	No Quality Assurance Plan (QAP) is available.	A Quality Assurance Plan (QAP) is available, but it is not consistently followed.	A Quality Assurance Plan (QAP) is available, approved, and consistently followed. The plan does not address resources, the planned resources are inadequate, or the planned resources are not available.	A Quality Assurance Plan (QAP) is available, approved, and consistently followed. The QAP identifies the necessary quality assurance resources, and those resources are available.	A Quality Assurance Plan (QAP) is available, approved, and consistently followed. The QAP identifies the necessary quality assurance resources, and those resources are available. QA has direct lines of communication to the Project Sponsor.
Quality Management	**Quality Metrics**	Quality data, including defects, is inconsistently tracked, and reported.	Quality metrics are limited to defects, but those are consistently tracked and reported.	Quality related metric data, including defects, is tracked, and reported.	Quality related metric data, including defects, is tracked, and reported with both point in them and trend data.	Quality related metric data, including defects, is tracked, compared with quality benchmark data, and reported with both point in them and trend data.

Category	KPA	Very Low	Low	Average	High	Very High
Quality Management	**Quality control**	Quality control is either non-existent or inconsistently applied.	Quality control focuses on the most critical documents, with no review of non-critical documents.	Deliverable document quality control and approval processes are comprehensive. Code quality control is based on testing.	Deliverable document quality control and approval processes are comprehensive. Code quality control is based on both testing and the use of automated code review tools.	Documents are reviewed for accuracy, completeness, suitability, and both internal and external consistency. Code walkthroughs plus automated code review tools are used for all code. Reviews include internal QA by the developing organization plus review by the government. Comment resolution matrices are used to track the resolution of all comments. Final approval includes the government Contract Officer, the Contract Officer's designated technical representative, and the Project Sponsor.
Quality Management	**Configuration Audits**	No configuration management is practiced.	No configuration management is enforced. Individuals and teams determine when and what to place under configuration control.	Configuration audits for the software are conducted.	Configuration audits verify that the team is working with the most recent version of documents and code.	Configuration audits are performed to maintain the integrity of configuration baselines, changes, and content of the configuration management system.

Category	KPA	Very Low	Low	Average	High	Very High
Quality Management	**Process Control**	Formal processes are not followed on the project.	Process control is focused on ensuring that key processes are in place.	Process control ensures that key processes are in place and followed.	Process control ensures that all significant processes are in place and followed.	A standard approach such as DMAIC (Define, Measure, Analyze, Improve, Control) is applied to process improvement.
Risk Management	**Risk Planning**	No Risk Management Plan exists.	A Risk Management Plan exists but it is neither approved nor consistently followed.	An approved Risk Management Plan is inconsistently followed.	An approved Risk Management Plan exists and is followed.	An approved Risk Management Plan exists and is followed. The Risk Management Plan includes suitable budgets for risk related activities, including risk mitigation activities.
Risk Management	**Risk Identification**	Risks are not formally identified or tracked.	Risk identification is the responsibility of a small number of people. Identified risks are assigned a unique ID and tracked to resolution in a risk register.	Risk identification is integral to all levels of the project, including all project meetings, plus it is a specific agenda item for status meetings. Identified risks are assigned a unique ID and tracked to resolution in a risk register.	Risk identification is integral to all levels of the project, including all project meetings, plus it is a specific agenda item for status meetings. Risks come from all stakeholders, both internal and external. Identified risks are assigned a unique ID and tracked to resolution in a risk register.	Risks include both threats and opportunities. Risk identification is integral to all levels of the project, including all project meetings, plus it is a specific agenda item for status meetings. Risks come from all stakeholders, both internal and external. Identified risks are assigned a unique ID and tracked to resolution in a risk register.

Category	KPA	Very Low	Low	Average	High	Very High
Risk Management	**Risk Categorization**	Risks sources and impacts are not consistently categorized.	Risks are somewhat categorized in terms of source and impact.	Risks are mostly categorized in terms of source and impact.	Risks are categorized in terms of source and impact, but the categorization uses an informal, non-standard taxonomy.	Risks are categorized by both the risk source/type (business/financial; health and safety; legal; project; quality; security; technical; other or multiple) and the impact category (impact to the project, impact to the organization, impact to external stakeholders).
Risk Management	**Risk Assessments**	Risks are not assessed in terms of exposure.	Risks are assessed using a single factor exposure number (e.g., Very Low to Very High).	Risks are assigned a qualitative probability (very low to very high), consequence (very low to very high), and overall exposure (very low to very high).	Risks are assigned a qualitative probability (very low to very high), consequence (very low to very high), and overall exposure (very low to very high). The most significant risks are analyzed using Delphi or other expert judgement-based approaches.	Risks are assigned a qualitative probability (very low to very high), consequence (very low to very high), and overall exposure (very low to very high). The most significant risks are analyzed using quantitative means (e.g., Monte Carlo simulation).

Category	KPA	Very Low	Low	Average	High	Very High
Risk Management	**Risk Management**	Formal risk management is not consistently applied.	Risk management is primary based on monitoring and responding.	For risks with high exposure, a risk management strategy is defined and implemented to decrease the risk exposure.	For risks with high exposure, a risk management strategy is defined and implemented to decrease the risk exposure. Where risks remain high following mitigation, an approved contingency plan is in place.	For risks with high exposure and for significant opportunities, a risk management strategy is defined and implemented to decrease the risk exposure for threats and increase the potential value of opportunities. Where risks remain high following mitigation, an approved contingency plan is in place.
Risk Management	**Risk Monitoring**	Risk monitoring is reactionary in nature, focusing on issues more than risks.	Regular meetings discuss outstanding risks to identify those that are of most concern at each given time.	Risk probabilities and impacts are updated on a regular basis. Risk monitoring is used to identify which risks become issues as part of regularly scheduled status meetings.	Risk probabilities and impacts are updated on a regular basis. Trigger events/criteria for risks are identified, and risk monitoring to identify which risks become issues is part of regularly scheduled status meetings.	Risk probabilities and impacts are updated on a regular basis. Trigger events/criteria for risks are identified, and risk monitoring to identify which risks become issues is part of regularly scheduled status meetings. Notification procedures are in place to notify relevant personnel, including the Project Sponsor, of risks that become issues or otherwise have significant changes.

Category	KPA	Very Low	Low	Average	High	Very High
Release Management	**Release Planning**	Release Planning is ad hoc, speculative, and informal. No Standardized release management process exists.	A release management plan exists, but it is significantly incomplete.	A complete release management plan exists, but it is both not approved and not maintained.	A complete release management plan exists, but it is either not approved or not maintained.	An approved release management plan describes the approach to performing release management. The plan includes activities, resources, roles, responsibilities, risks, criteria, and roll back strategy.
Release Management	**Release Documentation**	Release related documentation is informal and ad hoc.	Release documentation focuses on listing software changes.	Each release includes documentation tailored for that release, including a release test plan and release test report.	Each release includes documentation tailored for that release, including a release test plan and release test report. A version description document is prepared for each release, describing that version of the software, and including changes from previous versions.	Each release includes documentation tailored for that release, including a release test plan and release test report. A version description document is prepared for each release, describing that version of the software, and including changes from previous versions. Technical and user documentation is updated with each release to remain current.

Category	KPA	Very Low	Low	Average	High	Very High
Release Management	**Regression Testing**	Releases receive limited or no regression testing.	Release regression testing is fully or mostly manual and focuses on regression testing the portions of the system that have changed.	Release regression testing is partially automated and focuses on regression testing the portions of the system that have changed.	Releases include full regression testing using mostly automated process. ADA compliance testing is not part of the normal release cycle but is performed as needed.	Each release includes full regression testing and ADA compliance verification. Regression testing is fully automated using regression test scripts, and ADA compliance verification uses tool-based compliance validation.
Testing	**Unit Test Cases**	Unit testing is performed by the developers using ad hoc approaches.	Developers are responsible for creating their own unit test cases.	Unit test cases were prepared for use by the developers, but they do not explicitly include exception and boundary cases.	Comprehensive unit test cases were prepared for the developers, including exception and boundary conditions, but standardized test data was not created.	A comprehensive and approved set of unit test cases exists and was available to the developers prior to developing each functional area. A standardized set of unit test data was developed to accompany the test cases. Unit test cases include normal, exception, and boundary testing.
Testing	**System/ Integration Test Cases**	SIT testing is performed by the developers using ad hoc approaches	High level SIT test cases were available prior to starting SIT, but detailed testing activities are ad hoc.	A mostly comprehensive set of system/integration test (SIT) test cases was available prior to the start of SIT testing.	A comprehensive and approved set of system/integration test (SIT) test cases, along with appropriate test data, was available prior to the start of SIT testing.	A comprehensive and approved set of system/integration test (SIT) test cases, along with appropriate test data, was available prior to the start of SIT testing. SIT testing

Category	KPA	Very Low	Low	Average	High	Very High
						included interface testing using realistic load and boundary testing for interface capabilities.
Testing	**Performance Test Cases**	Performance testing is not included as part of the overall test strategy.	Performance testing is largely ad hoc and based on areas of the system that seem to run slow during other testing.	Performance test cases and data are available for the portions of the system that are expected to have the most significant performance problems.	A comprehensive and approved set of performance test cases was available prior to the completion of unit testing. Performance testing was supported either by a sanitized set of production data or a set of performance test data similar in size and characteristics to production data. Performance was tested with a 75% CPU reserve capacity.	A comprehensive and approved set of performance test cases was available prior to the completion of unit testing. Performance testing was supported either by a sanitized set of production data or a set of performance test data similar in size and characteristics to production data. Performance was tested with a 50% CPU reserve capacity.
Testing	**Functional Test Cases**	Functional testing is performed by the developers using ad hoc approaches.	Functional testing is performed by the SMEs using ad hoc approaches.	A comprehensive and approved set of functional test cases was available prior to the start of UAT.	A comprehensive and approved set of functional test cases was available prior to the start of UAT. Functional test data needed to fully test the system functionality, and	A comprehensive and approved set of functional test cases was available prior to the start of UAT. Functional test data needed to fully test the system functionality, and

Category	KPA	Very Low	Low	Average	High	Very High
					with known characteristics, is available.	with known characteristics, is available. Functional testing includes functional, boundary and exception testing.
Testing	**Regression Test Cases**	Regression testing is ad hoc, and the responsibility of individuals involved in other testing.	Regression test scripts and data are created as portions of the system are tested.	A full set of regression test cases, with standardized data, was available prior to the start of UAT.	A full set of regression test cases, with partial test automation, was available prior to the start of UAT.	A full set of regression test cases, with full test automation, was available prior to the start of UAT.
Testing	**User Acceptance Test Cases**	User acceptance testing is performed ad hoc by the developers with some involvement by the business users.	User acceptance testing is performed by the business users either ad hoc or using limited test cases.	A comprehensive and approved set of user acceptance test cases was available prior to the start of UAT. User acceptance test data needed to fully test the system capabilities, and with known characteristics, is available.	A comprehensive and approved set of user acceptance test cases was available prior to the start of UAT. User acceptance test data needed to fully test the system capabilities, and with known characteristics, is available. Users have the time and encouragement to intentionally try to "break" the system.	A comprehensive and approved set of user acceptance test cases was available prior to the start of UAT. User acceptance test data needed to fully test the system capabilities, and with known characteristics, is available. User acceptance testing includes functional, boundary and exception testing.

Category	KPA	Very Low	Low	Average	High	Very High
Testing	**Security Test Cases**	Security testing is not specifically addressed.	Security test cases are high level and directional in nature. An individual with computer security related training/certification is responsible for performing security testing, although the testing is largely ad hoc.	A partial set of security test cases was available prior to the start of SIT. Test cases focus on logical security, and cover most but not all of confidentiality, integrity, authentication, authorization, availability, and non-repudiation.	A comprehensive and approved set of security test cases was available prior to the start of SIT. Test cases cover all aspects of system security, both physical and logical. Test cases cover confidentiality, integrity, authentication, authorization, availability, and non-repudiation.	A comprehensive and approved set of security test cases was available prior to the start of SIT. Test cases cover all aspects of system security, both physical and logical. Test cases cover confidentiality, integrity, authentication, authorization, availability, and non-repudiation. A NIST compliant standard testing framework such as SCSEM is used.
Testing	**Test Readiness Review**	Feedback from the testing team was used to determine readiness for continued testing.	SIT and UAT testing schedules were fluid and phased, evolving as parts of the system were completed.	An informal review to confirm readiness for testing was conducted by the project team.	A TRR milestone review was conducted prior to the start of SIT and UAT, but this was not explicitly a gate review.	A TRR milestone gate review was conducted prior to the start of SIT and UAT.
Testing	**Production Readiness Review**	The decision to release to production is driven by a calendar date rather than system related criteria.	The decision to release to production is ad hoc or based on a single individual.	An informal review to confirm readiness for production was conducted by the project team.	A PRR milestone review was conducted prior to release to production, but this was not explicitly a gate review.	A PRR milestone gate review was conducted after UAT and prior to release to production.

Category	KPA	Very Low	Low	Average	High	Very High
Testing	**Test Planning**	Test planning is minimal, focused on timing for major test phases. Times allowed for testing are 10% or less of the overall development schedule.	Test planning is minimal, focused on timing for major test phases. Times allowed for testing a 20% or less of the overall development schedule.	Test planning is conducted but it is not comprehensive, and it is less formal. Often, it will consist of slides presented in a briefing.	A comprehensive Software Test Plan was prepared and followed, covering testing activities, schedule, roles, responsibilities, criteria, and resource requirements, and fully documenting all phases of testing from unit testing through release to production, but the plan was either not approved or not maintained.	A comprehensive and approved Software Test Plan was prepared and followed, covering testing activities, schedule, roles, responsibilities, criteria, and resource requirements, and fully documenting all phases of testing from unit testing through release to production. the plan is maintained to remain current.

Table 9 shows the assigned CARS project score for each of the eMRI capability categories. Justifications were included in Chapters Four, Five, and Six. We have also included a column showing the primary SOS organizational unit that would probably be responsible for implementing any improvements in each of the KPAs. In reading the CARS Score column:

- A score of Very Low or Low would be considered weak project capabilities, which will show up as some combination of risks to project success in terms of scope, budget, schedule, quality, technical objectives, and total cost of ownership. These projects tend to be unpredictable, and they have a high failure rate.

- A score of Very High or High would be considered a strong project score, which will show up as successful projects in terms of scope, budget, schedule, quality, achieving technical objectives, and total cost of ownership. In this context, success involves setting realistic objectives and then achieving those objectives. Strong project process skills result in predictability, not necessarily “cheap” projects in terms of cost.

- A score of Average would be neutral, with some good characteristics and some areas for improvement.

Table 9: CARS eMRI assessment scores.

Resp.	Category	KPA	CARS Score
PRD	Data Conversion and Migration	Data Integrity	Very Low
PRD	Data Conversion and Migration	Data Quality	Low
PRD	Data Conversion and Migration	Data Control	Very Low
PRD	Data Conversion and Migration	Data Security	Very Low
PRD	Data Conversion and Migration	Conversion Planning	Low
PRD	Data Conversion and Migration	Conversion Specifications	Very Low
PRD	Data Conversion and Migration	Conversion Architecture	Average
PRD	Data Conversion and Migration	Conversion Scheduling	Very Low
PRD	Data Conversion and Migration	Conversion Cutover	Very Low
PMO	Contract Management and Vendor Negotiations	Training	Very Low
PMO	Contract Management and Vendor Negotiations	References	Very Low
PMO	Contract Management and Vendor Negotiations	Vendor Evaluations	Very Low
PMO	Contract Management and Vendor Negotiations	Contract Negotiations	Low
PMO	Contract Management and Vendor Negotiations	Contract Change Management	Low
PMO	Contract Management and Vendor Negotiations	Competition	Very Low
PMO	Contract Management and Vendor Negotiations	Contract Risk Allocation	Very Low
PMO	Contract Management and Vendor Negotiations	Deliverable Management	Average
PRD	Requirement Definition and Management	Functional Requirements	High
PRD	Requirement Definition and Management	Non-Functional Requirements	High
PRD	Requirement Definition and Management	Requirement Structure	High
PRD	Requirement Definition and Management	Requirement Storage	Average
PRD	Requirement Definition and Management	Traceability	High
PRD	Requirement Definition and Management	Requirement Status	Very High
PRD	Requirement Definition and Management	Requirement Management Processes	Average
PRD	Requirement Definition and Management	Business Process Improvement	High
PMO	Project Schedule Management	Integrated Project Scheduling	High
PMO	Project Schedule Management	Milestone Reviews	Low
PMO	Project Schedule Management	Schedule Updates	Average
PMO	Project Schedule Management	Project Status	Average
PMO	Project Schedule Management	Estimating	Very Low
PMO	Project Schedule Management	Resourcing	Low
PMO	Communication Management	Communications Strategy	Low
PMO	Communication Management	Communication Planning	Average
PMO	Communication Management	Stakeholder Definition	High

Resp.	Category	KPA	CARS Score
PMO	Communication Management	Project Status Communication	Low
PMO	Communication Management	Action Item Management	Low
PMO	Communication Management	Meeting Management	Average
PRD	Governance and Sponsorship	Governance Structure	Low
PRD	Governance and Sponsorship	Expectations Management	Average
PRD	Governance and Sponsorship	Project Charter	Average
PRD	Governance and Sponsorship	Project Controls and Monitoring	Low
PRD	Organizational Change Management	OCM Planning	Very High
PRD	Organizational Change Management	OCM Methodology	High
PRD	Organizational Change Management	Gap Analysis	Low
PRD	Organizational Change Management	Training Plan	Low
PRD	Organizational Change Management	Training Records	Very Low
PMO	Quality Management	Configuration Control	Very Low
PMO	Quality Management	Process Documentation	Low
PMO	Quality Management	Quality Planning	Low
PMO	Quality Management	Quality Metrics	Low
PMO	Quality Management	Quality control	Low
PMO	Quality Management	Configuration Audits	Low
PMO	Quality Management	Process Control	Low
PMO	Risk Management	Risk Planning	High
PMO	Risk Management	Risk Identification	Average
PMO	Risk Management	Risk Categorization	Very Low
PMO	Risk Management	Risk Assessments	Low
PMO	Risk Management	Risk Management	Low
PMO	Risk Management	Risk Monitoring	Average
ITD	Release Management	Release Planning	Very Low
ITD	Release Management	Release Documentation	Low
ITD	Release Management	Regression Testing	Average
PRD	Testing	Unit Test Cases	Low
PRD	Testing	System/Integration Test Cases	Low
PRD	Testing	Performance Test Cases	Average
PRD	Testing	Functional Test Cases	Low
PRD	Testing	Regression Test Cases	Low
PRD	Testing	User Acceptance Test Cases	Low
PRD	Testing	Security Test Cases	Very Low

Resp.	Category	KPA	CARS Score
PRD	Testing	Test Readiness Review	Very Low
PRD	Testing	Production Readiness Review	Average
PRD	Testing	Test Planning	Average