
VA VistA Metadata Program Management Plan (PMP)

January 8, 2016

Version 1.0

To:

Mr. Alan Furuno, COR
1300 North Holocono Street
Suite U-216
Kihei, HI 96753
(808) 433-3602
alan.furuno@pacifichui.org

Submitted by:

Hokukahu, LLC
1050 Bishop Street, Suite 472
Honolulu, HI 96813
Primary Contact: Renton Nip
(808) 927-0999
rnip@hawaiiirg.com



Hokukahu, LLC

SBA-Certified 8(a) Company Majority Owned by Hokupili Foundation a native Hawaiian Organization

CONTENTS

1	Introduction.....	4
1.1	Project Description	4
2	Stakeholders.....	5
3	Program Management Overview.....	5
3.1	Recurring PM Activities	5
4	Management Controls.....	6
4.1	Technical Requirements Management	6
4.2	Agile Methodology.....	6
4.2.1	Scrum Overview.....	7
4.2.2	Agile Execution Process & Development Sprints	7
4.2.3	Sprint Planning Meetings.....	7
4.2.4	Weekly Project Meetings.....	7
4.3	Quality Assurance.....	8
5	Risk Identification and Management	8
6	Cost Requirements.....	9
7	Staffing Plan.....	9
8	Schedule Management Approach.....	10
8.1	Project Schedule (includes Work Breakdown Schedule)	14

LIST OF TABLES

Table 1: Project Staffing Plan.....	9
Table 2: Project Deliverable Tracks	10
Table 3: Metadata Definitions and System Configurations	11
Table 4: Software Deliverables.....	11
Table 5: Technical Document Deliverables	13
Table 6: Project Schedule for Program Management Deliverables	14
Table 7: Project Schedule for Technical Deliverables	15

LIST OF FIGURES

Figure 1: The Sprint Cycle	7
Figure 2: Risk Matrix	8

1 Introduction

This document presents the Program Management Plan (PMP) that the management team will implement throughout the VA VistA Metadata Project (hereafter, “the Project”). Hokukahu, LLC (hereafter, “Hokukahu”) will provide program management (PM) support to manage the cost, schedule, and performance of the Project’s Performance Work Statement (PWS) in accordance with industry best practices. This document outlines the overall plan/strategy to accomplish the tasks outlined in the PWS.

1.1 Project Description

The Project will provide a single comprehensive security enabled read/write data model for all VA VISTA data across all VA VISTA operational systems, establishing a common technical foundation for master data management and computable data representation and exchanging between VA and DoD clinical information systems. All artifacts and deliverables shall be developed, version-controlled, stored, and delivered on an industry-standard public GitHub repository.

Key capabilities of the Project include:

- Provide comprehensive always-up-to-date, machine-processable exposure and definition of complete operational VISTA data model (VDM), based on all data dictionaries from all active VISTA instances in standard machine-processable, exchangeable form, supported by off-the-shelf tools.
- Create a fully audited and normalized VISTA data model (MVDM) with no redundancy.
- Modify FileMan data to allow management (query, security, and read/write) of Patient, Institutional, Knowledge, and Systems data as distinct entities and to enable patient-centric security.

Functional benefits of the Project include:

- Provide comprehensive always-up-to-date, machine-processable exposure and definition of complete operational VISTA data model (VDM), based on all data dictionaries from all active VISTA instances in standard machine-processable, exchangeable form, supported by off-the-shelf tools.
- Create a fully audited and normalized VISTA data model (MVDM) with no redundancy.
- Modify FileMan data to allow management (query, security, and read/write) of Patient, Institutional, Knowledge, and Systems data as distinct entities and to enable patient-centric security.

2 Stakeholders

The key stakeholders for the Project are:

- PMO Lead: CAPT Paul Miller
- Program Manager: Renton Nip, Hokukahu
- Functional Sponsor: Mark Goodge
- Functional Proponents (VA): Rafael Richards, MD

3 Program Management Overview

Hokukahu will provide program management (PM) support to manage the cost, schedule, and performance of the PWS in accordance with industry best practices. The Project's Program Manager will be responsible for providing the status reports and other associated PM documents, and will ensure that PM activities, tasks, milestones, documentation and artifacts, will be consolidated and communicated as required.

3.1 Recurring PM Activities

Hokukahu will provide the following recurring program management activities:

- Develop, manage, and maintain a Project Schedule, and provide Monthly Updates;
- Ensure that all PWS requirements are performed and deliverables are submitted per agreed upon Project Schedule;
- Communicate regularly and effectively with the CO, Pacific JITC leadership and Stakeholder(s);
- Effectively balance and schedule involvement of appropriate technical resources throughout the project lifecycle;
- Implement, manage, and execute a risk management process;
- Provide project status reporting in monthly progress report;
- Develop and publish all artifacts, code, data, documentation, progress reports, and deliverables on single authoritative Project Repository (defined in section 2.1.8 of the PWS) and provide unlimited access to this for all project management activities;
- Provide a quarterly strategic communications message regarding project progress and feasibility of transition to production; and

- Perform financial management reporting, to be delivered with the monthly progress reports. Financial reporting shall include costs to date, planned costs, variance of spend to plan and identification of any funding issues.

4 Management Controls

Weekly meetings will be held by the Hokukahu Team to review task and technical status, schedule integration, testing, and demonstration activities, and review quality/client issues, and identify and mitigate risks.

The Program Manager will meet with the Contracting Officer's Representative (COR) as may be requested to review Project status and action items, coordinate government and Contractor activities, and obtain government guidance and direction.

Hokukahu will host progress meetings with the stakeholders at COR direction. This will ensure close coordination between Hokukahu and the stakeholders.

4.1 Technical Requirements Management

The Hokukahu Team will manage requirements through requirement analysis and technical iteration with the Government to ensure a thorough mutual understanding of the requirements and efforts. The Hokukahu Team's overarching objective is to provide high value to the Government by understanding and meeting the stated requirements of the Project.

The Hokukahu Team will iterate technical status with the Government through weekly technical meetings or more often if necessary. The purpose of this iterative interaction is to provide updated status information and to allow an opportunity for discussions relating to requirements, requirement modifications, and the overall technical direction of the Project.

As the PWS requires, all artifacts, code, data, documentation, progress reports, and deliverables will be published by the Team on a public GitHub (the Project Repository). This will provide unlimited access by Government and all stakeholders for all project management activities.

For the duration of the Project, a Scrum process will be in place for the research and prototyping process. Sprint summaries and access to GitHub will provide comprehensive project visibility to stakeholders. The Project Manager will handle internal communication with the Hokukahu Program Manager and the research and technical teams.

4.2 Agile Methodology

The Hokukahu Team will undertake the research and prototyping tasks of the project utilizing the Agile Management methodology and the scrum process. Scrum enables the planning, research, and technical team to be more responsive, flexible, and adaptable by implementing iterations within the project task cycle.

4.2.1 Scrum Overview

In a scrum framework, progress is made through incremental sprints with a limited duration of one to four weeks. In addition, the scrum will provide project planning, design, and development visibility to stakeholders. The scrum framework also encourages a multi-directional flow of information and guidance between the Government, subcontractors, project management, and the technical team.

4.2.2 Agile Execution Process & Development Sprints

The research analysis and prototypes of the project will consist of a series of planned sprints. At the beginning of the Contract, the project requirements will be broken down into prioritized items. Each sprint will be planned to deliver features and analysis related to the subsystem(s) or components in focus.

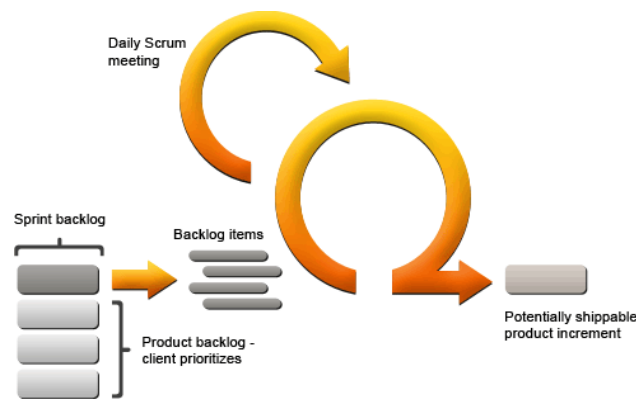


Figure 1: The Sprint Cycle

4.2.3 Sprint Planning Meetings

A sprint planning meeting will be held before each sprint to identify the focus of the next sprint. The Project Manager will assist the research and development teams and work to remove or mitigate any possible impediments in order to ensure that the teams have the best possible circumstances for realizing the goals fixed for the sprint. Each sprint will enhance the research analysis and outcome, and prototype value and will add new functions and improvements that will be delivered.

4.2.4 Weekly Project Meetings

Weekly project meetings will be held with the entire team to review tasks and the status of sprints, track Contract deliverables, schedule integration, testing and demonstration activities, and review quality/client issues.

4.3 Quality Assurance

The purpose of Quality Assurance (QA) is to provide staff and management with objective insight into processes and associated work products.

QA supports the delivery of high-quality products and services by providing the staff and managers at all levels with appropriate visibility into, and feedback on, processes and associated work products throughout the life of the project. QA involves objectively evaluating work products against applicable quality standards, practices and procedures, in addition to objectively auditing process compliance against the applicable process descriptions, standards, and procedures.

The Scrum will build QA into each stage of the cycle and sprints. QA best practices will also apply to all research. Testing will be performed within engineering cycles. QA testing, including but not limited to bugs and bug resolution will be tracked.

5 Risk Identification and Management

The Hokukahu Team will identify and assess risks, as well as potential resolutions and related costs throughout the duration of the project.

Risks associated with the Project are those aspects of the proposal the Team considers to have the potential for disruption of schedule, poor performance, the need for increased Government oversight, and/or the likelihood of unsuccessful contract performance.

The Team will report any perceived risk related to the Project to the Government during quarterly update meetings (following the risk assessment schema made available in ProSight), or as directed by the Government. Risk will be defined as anything that will have a negative impact on any one or all of the primary project constraints - Time, Resources and Performance Criteria. A risk matrix will be populated with Risk Score(s). Where appropriate, the Team will provide mitigating strategies for reported risks.

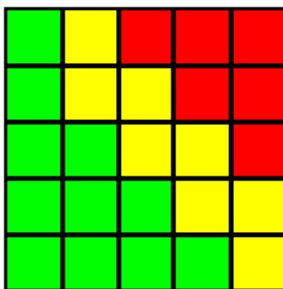


Figure 2: Risk Matrix

For each risk, a Risk Score will be a calculated value based on an assessment of Risk Likelihood (y-axis) and Consequence (x-axis) with an individual value from 1 (low) to 5 (high) for each.

6 Cost Requirements

The cost requirements for the Project were fully proposed and determined in the contracting process. The resultant Contract is a firm fixed price award and thus requires no cost monitoring by the Government. Hokukahu, LLC has adequate resources to ensure performance even if the award amount is inadequate to cover actual project expenses. Hokukahu, LLC will of course be monitoring its costs in relation to the Project. Its fiscal data is proprietary.

7 Staffing Plan

The Hokukahu Team will include the personnel summarized in the following tables. The level of effort for one full time employee (FTE) is estimated at 1920 hours. Additional information on labor categories for this Project is provided in the Cost Proposal.

Table 1: Project Staffing Plan

Labor Category	FTE over PoP	Function
Program/Contract Manager	.5	Organizes, directs, and manages contract operation and personnel
Subject Matter Expert (Master)	3	Serves as SME, possessing in-depth knowledge of highly specialized applications/systems.
Applications Developer (Master)	4	Designs, develops, enhances, debugs, and implements software or major enhancements to existing software.
Applications Developer (Senior)	3	Designs, develops, enhances, debugs, and implements software or major enhancements to existing software.
Test Engineer (Senior)	.5	Evaluates and implements test tools and strategies. Designs and conducts test and evaluation procedures.
Technical Writer (Senior)	1	Produces and reviews content of technical documentation as appropriate to the requirements.
Project Manager	1	Provides Project management and scrum coordination roles.

8 Schedule Management Approach

In order to develop a Project Schedule and a Work Breakdown Schedule (WBS), the Team undertook a review of Project requirements, in particular, the Project deliverables. Upon analysis, the Team organized related deliverables into five functional tracks. In the Project Repository, each track will be supported by a separate Git. The Tracks are described in the following table.

Table 2: Project Deliverable Tracks

Track	Name	Description	GIT	Technical Deliverable
A	Infrastructure	Project infrastructure including Test VISTA ("nodeVISTA"), gits, tooling, website	documents, nodeVISTA, website	3
B	VDM	VISTA Data Model (VDM) - native model exposure and package implementation	VDM	12
C	MVDM	Master VISTA Data Model (MVDM) - definition and implementation of master data model for VISTA	MVDM	9
D	MVDMmap	Mapping MVDM to other models such as FHIR	MVDMmap	3
PM	Project Management	Business/Project Management Documents	documents	N/A

The following tables (Table 3: Metadata Definitions and System Configurations, Table 4: Software Deliverables, and Table 5: Technical Document Deliverables) specify the distinct artifacts that were identified by the Team after analyzing the requirements and deliverables.

Table 4: Software Deliverables), in addition to the deliverables listed in the Project's PWS (Section 8.2), identifies additional deliverables for planning purposes that are necessary for the development of PWS deliverables. Such deliverables have been identified with a prefix of "E". Deliverables 7, 10, and 11 were divided and designated .1 and .2 for VDM and MVDM, respectively.

Table 3: Metadata Definitions and System Configurations

#	Name	Format	Function	Deliverable(s)
1	dd.jsonld	JSON-LD	Formal, portable definition of the contents of a VISTA data dictionary	8
2	rpc.jsonld	JSON-LD	Formal definition of the model implicit in RPCs, captured in JSON-LD	E1
3	vpr.jsonld	JSON-LD	Formal definition of the VPR RPC's patient data model in JSON-LD	Part of 10.1
4	vdm.jsonld	JSON-LD	Formal definition of Native VISTA data model based on one or more dd.jsonld's and rpc.jsonld	7.1, 7.2
5	mvdm.jsonld	JSON-LD	Formal definition of the MVDM subset of VDM that supports full CRUD (Create, Read, Update and Delete)	10.1, 10.2
6	piks.jsonld	JSON-LD	Formal annotation of vdm.jsonld that distinguishes Patient, Institution, Knowledge and System (PIKS) classes and properties	18
7	nodeVISTA Scenarios	GT.M and Cache Databases	VISTA databases for testing and demonstrations	Part of E2.2
8	MVDM to FHIR Rules	Rules Format	Translation rules (MVDM to FHIR)	Part of 39

Table 4: Software Deliverables

#	Name	Function	Deliverable(s)
1	DDJLD Maker	Caches FileMan Data Dictionary (dd) from a VISTA and creates a dd.jsonld	8
2	RPCJLD Maker	Caches RPC definitions from a VISTA and creates anrpc.jsonld	E1

#	Name	Function	Deliverable(s)
3	nodeVISTA	A test VISTA based on OSEHRA's VISTA and a simple node.js front end	E3
4	nodeVISTA Commands	invocations of mainly write-back functions in VISTA to prepare for the write-back support of VDM Package	Part of 7.2, E2.2
5	VDM Maker	Creates a VISTA Data Model (VDM), vdm.jsonld, from a VISTA's dd.jsonld and rpc.jsonld	7.1, 7.2
6	VDM Package	Implements VDM inside FileMan. The first version will support querying ("Read-only"). The full version will support Create-Read-Update-Delete and transactions.	E2.1, E2.2
7	MVDM Maker	Creates a Master VISTA Data Model (MVDM), mvdm.jsonld, from one or more vdm.jsonld's and vpr.jsonld	10.1, 10.2
8	MVDM Module	Implements MVDM inside FileMan over the VDM Package. The first version will support querying ("Read-only"). The full version will support Create-Read-Update-Delete and transactions.	11.1, 11.2
9	MVDM Test Suite	A series of tests focused on write-back support of the MVDM Module. Tests VDM write-back as MVDM relies on VDM.	35
10	PIKS Generator	Generates Patient, Institution, Knowledge and System (PIKS) annotations in piks.jsonld for a vdm.jsonld	19
11	Patient Security Prototype	An illustration of PIKS-enabled Patient level security. This involves an example client and an addition to FQS	28
12	FQS	FileMan Query Service (FQS) based on embedded VDM model (REST service; read only)	25
13	Example Query Clients	Example command line clients that show how to use the FQS	25
14	FQS Web Client	Browser based client for using the FQS	33

#	Name	Function	Deliverable(s)
15	Metadata Cacher	queries (VISTA Application) metadata using VDM Package	15
16	MVDMmap Prototypes	prototypes showing mapping from VDMN to FHIR	39
17	Web-based Rules Hub	host for Translation rules	32
18	Document Generators	Generators of documentation leveraging common packages such as Sphinx and JSDoc and translators from Markdown to PDF and HTML	E4

Table 5: Technical Document Deliverables

#	Name	Deliverable(s)
1	Website	13
2	(Document) Approach to “Live VDM” Maintenance of Current State	9
3	[MVDM] Normalization Reports	12
4	Report on [MVDM] Exposure of older models	14
5	Prototype Patient-centric Data Security [Document]	28 (Document)
6	Document VISTA-ese vs. FHIR	40

Through the above-described analysis, the Team developed the Project Schedule that appears in the following section. The Project Schedule is a living document that will evolve throughout the Project. Potential revisions to the Project Schedule will be an agenda item in the weekly meetings with the Government.

8.1 Project Schedule (includes Work Breakdown Schedule)

Utilizing the schedule management approach described in the previous section, the Team developed a Project Schedule to include a Work Breakdown Schedule (WBS):

Table 6: Project Schedule for Program Management Deliverables

#	Track	Name	WBS	Due	Content	PWS
1A	PM	Non-disclosure/Non-Use Agreement	Q1	Prior to Work Performance		6.1
1B	PM	Quality Control Plan [QCP]	Q1	30 days after contract award	An effective quality control program	1.6.1
1C	PM	Phase-out Migration Plan	Q4	30 days prior to end of contract	Elaborates the artifacts to be transitioned on the Project Repository, and a schedule for transition completion	1.6.17
2	PM	Program Management Plan (PMP)	Q1	1/8/2016	Strategy to accomplish the tasks and include the risk, quality and technical management approach, WBS, schedule management approach, schedule, cost requirements, and proposed staffing	5.2
3	PM	Program Schedule and Monthly Updates	Monthly	Monthly	Schedule, updated monthly	5.2
4	PM	Monthly Progress Report	Monthly	Monthly	Project status, financial management reporting	5.4.2
5	PM	Quarterly Strategic Communications Message	Quarterly	Quarterly	Project progress, feasibility of transition to production	5.2

Table 7: Project Schedule for Technical Deliverables

#	Track	Name	WBS	Due	Content	PWS
1AA	A	Artifact Repository	Q1	1/8/2016	Project Gits	8.2
13	A	Website	Q1 > Q4	Final version at contract completion date.	website, infographics to showcase the contents of the VDM and MVDM Subset	5.3.2
E3	A	FileMan TEST VISTA ["nodeVISTA"]	Q1 > Q4	Not a contract deliverable.	a test VISTA ("nodeVISTA") that hosts different test datasets ("nodeVISTA Scenarios")	
E4	A	Document Generator	Q1 > Q3	Not a contract deliverable.	Programmer documentation will be generated using tools such as Sphinx and JSDoc	
7.1	B	Machine Processable VISTA Data Model (VDM) "Read-only"	Q1	Final version at contract completion date.	vdm.jsonld, the native VISTA data model in JSON-LD based on one or more dd.jsonld's. VDM Maker, a program that creates vdm.jsonld from dd.jsonld's. This version will support query/read ("VDM (read)").	5.3.1
7.2	B	Machine Processable VISTA Data Model (VDM)	Q2 > Q4	Final version at contract completion date.	vdm.jsonld, enhanced by write-data in dd.jsonlds and rpc.jsonld. <i>VDM Maker</i> must process more information from dd.jsonld's and process rpc.jsonld.	5.3.1

#	Track	Name	WBS	Due	Content	PWS
8	B	Date-stamped FileMan Data Model Implementations (Definitions) (cross refs, triggers ...)	Q1 > Q2	Final version at contract completion date.	dd.jsonld, a data dictionary captured in JSON-LD. DDJLD Maker, a program that caches and interprets the dictionaries from VISTAs in JSON-LD form. MUMPS code reduction will be needed for write-back support Required for #7.	5.3.1
E1	B	RPC Model	Q1 > Q3	Not a contract deliverable.	formal definition of the model implicit in RPCs (rpc.jsonld) for vdm.jsonld (#7)	
E2.1	B	VDM Package "Read-only"	Q1	Not a contract deliverable.	a package that implements the VDM inside a VISTA. It will allow any FileMan data to be queried according to the VDM.	
E2.2	B	VDM Package	Q1 > Q4	Not a contract deliverable.	Will add support for creating, updating and deleting (full CRUD) VISTA Data according to a write-back supporting VDM (#7.2). Initial write-back testing will be directly against nodeVISTA ("nodeVISTA Commands")	
9	B	(Document) Approach to "Live VDM" Maintenance of Current State	Q4	Final version at contract completion date.	In a wiki page, describe ways in which dd.jsonld definitions and hence vdm.jsonld could keep pace with changes in VISTAs	5.3.1

#	Track	Name	WBS	Due	Content	PWS
15	B	Date Stamped (Application) Meta Data for lab, surgery and other applications	Q2	Final version at contract completion date.	<i>Metadata Cacher</i> that queries meta-data using the (read-only) <i>VDM package</i> .	5.3.3
18	B	Machine-processable [PIKS] Annotations	Q2	Final version at contract completion date.	Distinguish patient data from other types of VISTA data. VDM PIKS enables MVDM PIKS which enables patient-centric security (#28)	5.3.4
19	B	Software code [for PIKS]	Q2	Final version at contract completion date.	PIKS Annotation Generator	5.3.4
25	B	Prototype query access to VISTA Data against VDM ["FQS"]	Q2	Final version at contract completion date.	Example clients that query (read-only) the FileMan Test VISTA using a REST-based FileMan Query Service (FQS) implemented over a read-only version of the VDM Package	5.4.1
33	B	Prototype Web-Based Query Interface to FileMan [VDM] Data	Q2 > Q3	Final version at contract completion date.	Simple Web Client for using a read-only version of the VDM Package	5.4.1
10.1	C	Master VISTA Data Model (MVDM) "Read-only"	Q1 > Q2	Final version at contract completion date.	mvdm.jsonld, a formal "MVDM Subset" definition with much of the scope of the VPR RPC which must be formally captured in _vpr.jsonld_	5.3.2
10.2	C	Master VISTA Data Model (MVDM)	Q2 > Q4	Final version at contract completion date.	Full CRUD support rounded out for mvdm.jsonld.	5.3.2

#	Track	Name	WBS	Due	Content	PWS
11.1	C	[MVDM over VDM] Heuristic (mapping) code "Read-only"	Q2	Final version at contract completion date.	Mapping tables and rules implemented in a MVDM module that delivers a read-only version of MVDM over the VDM Package "Read-only".	5.3.2
11.2	C	[MVDM over VDM] Heuristic (mapping) code	Q3 > Q4	Final version at contract completion date.	Full CRUD support added to "Read-only" base.	5.3.2
12	C	[MVDM] Normalization Reports	Q2 > Q4	Final version at contract completion date.	Documents VDM to MVDM mapping to be implemented in #11	5.3.2
14	C	Report on [MVDM] Exposure of older models	Q4	Final version at contract completion date.	Describe how older, cruder models could be handled in the MVDM	5.3.2
28	C	Prototype Patient-centric Data Security	Q3 > Q4	Final version at contract completion date.	First document and then provide a self- contained prototype that shows how PIKS- enabled annotations enable patient-centric secure queries	5.4.1
35	C	VISTA Application model(s)/Prototype(s) [Tests]	Q2 > Q4	Final version at contract completion date.	MVDM write back tests, enabled by vdmn.js configurations. Test scenarios for Deliverable #11.	5.4.2
36	C	Meta-model(s) [VPR] Prototype(s)	Q2 > Q3	Final version at contract completion date.	Test code that shows how well the MVDM supports VPR (Read-Only) convenience methods. Read-only side of #35.	5.4.2

#	Track	Name	WBS	Due	Content	PWS
32	D	Prototype Web-based Rules Hub	Q3	Final version at contract completion date.	Prototype a sharable, crowd source-able mechanism to exchange and grow a library of open, standards-based, validated, and exchangeable transformation rules	5.4.1
39	D	Reference models(s)/Prototype(s)	Q3 > Q4	Final version at contract completion date.	Prototype that demonstrates a mapping from MVDM to FHIR	5.4.2
40	D	Document VISTA-ese vs. FHIR	Q3 > Q4	Final version at contract completion date.	Human-readable mapping descriptions	5.4.2

The WBS is more specifically delineated in a separate attachment provided that accompanies this submission.

The Project Schedule is a subject to Government approval. Potential revisions to the Project Schedule will be an agenda item in the weekly meetings with the Government.