NASA

National Aeronautics and Space Administration

Office of Inspector General

Office of Audits

AUDIT OF THE SPACEPORT COMMAND AND CONTROL SYSTEM

March 28, 2016

Report No. IG-16-015

Office of Inspector General

RESULTS IN BRIEF

Audit of the Spaceport Command and Control System

March 28, 2016 IG-16-015 (A-15-008-00)

WHY WE PERFORMED THIS AUDIT

NASA's Ground Systems Development and Operations (GSDO) Program is responsible for preparing the Kennedy Space Center (Kennedy) to launch the next generation of rockets and spacecraft, including the Space Launch System (SLS) and Orion Multi-Purpose Crew Vehicle (Orion) NASA plans to use for deep space exploration. To accomplish this mission, the GSDO Program must move vehicles to launch pads, manage and operate the equipment required to connect spacecraft with rockets, and send the integrated vehicles into space. As part of this effort, the GSDO Program is developing the Spaceport Command and Control System (SCCS) – a software system that will control pumps, motors, valves, power supplies, and other ground equipment; record and retrieve data from systems before and during launch; and monitor the health and status of spacecraft as they prepare for and launch. To create the SCCS, NASA is writing a large amount of computer code to "glue" together multiple existing software products or, in some cases, the parts of those products the Agency deems most effective for its purposes.

In the past, NASA has experienced difficulties with similar large, complex software development efforts. For example, between 1995 and 2002, the Agency spent more than $500 million on two separate attempts to update command and control software at Kennedy. Both efforts failed to meet their objectives and were substantially scaled back or cancelled prior to completion.

In this audit we examined whether NASA is effectively managing the SCCS software development effort. To complete this objective, we performed work at Kennedy, interviewed GSDO Program officials and commercial companies involved with command and control software, and reviewed various studies concerning the SCCS, Federal laws, and NASA policies.

WHAT WE FOUND

The SCCS development effort has significantly exceeded initial cost and schedule estimates. Compared to fiscal year 2012 projections, development costs have increased approximately 77 percent to $207.4 million and the release of a fully operational version has slipped by 14 months from July 2016 to September 2017. In addition, several planned capabilities have been deferred because of cost and timing pressures, including the ability to automatically detect the root cause of specific equipment and system failures. Without this information, it will be more difficult for controllers and engineers to quickly diagnose and resolve issues. Although NASA officials believe the SCCS will operate safely without these capabilities, they acknowledge the reduced capability could affect the ability to react to unexpected issues during launch operations and potentially impact the launch schedule for the combined SLS-Orion system.

The root of these issues largely results from NASA's implementation of its June 2006 decision to integrate multiple products or, in some cases, parts of products rather than developing software in-house or buying an off-the-shelf product. Writing computer code to "glue" together disparate products has turned out to be more complex and expensive than anticipated. As of January 2016, Agency personnel had developed 2.5 million lines of "glue-ware," with almost two more years of development activity planned. In comparison, NASA reengineered the Hubble Space Telescope command and control system with approximately 500,000 lines of "glue-ware" code.

SCCS Project managers told us the decision to develop SCCS in this manner was motivated by several factors. Managers did not want to rely on a single company's software because if that company encountered financial difficulties or stopped providing technical support NASA's space exploration efforts could be negatively impacted. In addition, at the time the decision was made, managers believed the effort to integrate the various software products would not be overly time-consuming or technically complex. While that decision may have been reasonable based on what managers knew at the time, it is now clear they underestimated the complexity of the software integration activities that would be required.

In the past, NASA has encountered difficulties with large and complex command and control software development efforts, failing on two occasions to meet expected requirements despite spending more than $500 million. In something of a repeat of this pattern, the SCCS development effort is more than 1 year behind schedule and significantly over cost, and several planned software capabilities have been deferred.

NASA made its decision regarding the SCCS software architecture nearly 10 years ago, but in our view this may no longer be the most prudent course of action given significant advances in commercial command and control software over that time. For example, the two companies under contract with NASA to deliver supplies to the International Space Station – Orbital Sciences Corporation and Space Exploration Technologies – both use commercial software products to accomplish their missions. In our judgment, the GSDO Program's reluctance to change course reflects a cultural legacy at NASA of over-optimism and over-promising what the Agency can achieve in a specific timeframe.

What We Recommended

In a draft of this report, we recommended the Associate Administrator for Human Exploration and Operations commission an independent assessment to evaluate the status of the SCCS software development effort and determine the necessary steps to reduce the risk of further cost, schedule, and performance issues, including consideration of acquiring commercial command and control software to replace some or all of the system currently under development.

NASA agreed to conduct an independent assessment of the command and control system once software for Exploration Mission-1 – the first launch of the combined SLS-Orion system scheduled for November 2018 – is successfully delivered. We consider management's plan responsive to our recommendation. Therefore, the recommendation is resolved and will be closed upon completion and verification of the proposed corrective action.

For more information on the NASA Office of Inspector General and to view this and other reports visit https://oig.nasa.gov/.

TABLE OF CONTENTS

Acronyms

ATO	Authorization to Operate
CMMI	Capability Maturity Model Integration
COTS	Commercial Off-The-Shelf
FY	Fiscal Year
GAO	Government Accountability Office
GSDO	Ground Systems Development and Operations
NIST	National Institute of Standards and Technology
NPR	NASA Procedural Requirements
OIG	Office of Inspector General
PPBE	Planning, Programming, Budgeting, and Execution
SCCS	Spaceport Command and Control System
SLS	Space Launch System

INTRODUCTION

The Ground Systems Development and Operations (GSDO) Program is responsible for preparing the Kennedy Space Center (Kennedy) to launch the next generation of rockets and spacecraft, including the Space Launch System (SLS) and Orion Multi-Purpose Crew Vehicle (Orion), which NASA plans to use for deep space exploration. To accomplish this mission, the GSDO Program must move vehicles to launch pads, manage and operate the equipment required to connect spacecraft with rockets safely, and successfully send the integrated vehicles into space. As part of this effort, the GSDO Program is developing the Spaceport Command and Control System (SCCS), a software system that will control pumps, motors, valves, power supplies, and other ground equipment; record and retrieve data from a variety of systems before and during launch; and monitor the health and status of spacecraft as they prepare for and launch. NASA is writing a large amount of computer code that will "glue" together multiple existing software products or, in some cases, the parts of those products the Agency deems most effective for its purposes to create the SCCS. The Agency estimates that its efforts to develop and certify the SCCS will cost as much as $207.4 million.

In the past, NASA has experienced difficulties with similar large, complex software development efforts. For example, between 1995 and 2002, the Agency spent more than $500 million on two separate attempts to update command and control software at Kennedy. Unfortunately, both efforts failed to meet their objectives and were substantially scaled back or cancelled prior to completion. As we indicated in a March 2015 report, the SCCS is facing similar cost and schedule issues.[1] Moreover, in its 2014 Annual Report, NASA's Aerospace Safety Advisory Panel expressed concern about the SCCS development effort, noting schedule slips and difficulties obtaining necessary software certification.[2]

In this audit, we examined whether NASA is effectively managing the SCCS development effort, a critical part of the Agency's plans to launch the SLS rocket and Orion capsule in the next 7 years. Details of the audit's scope and methodology are outlined in Appendix A.

Background

Software encompasses computer programs, procedures, rules, and the associated documentation and data pertaining to the development and operation of a computer system. Ensuring the quality and reliability of software used in NASA missions is essential to achieving mission success. NASA policy dictates a disciplined approach to acquire, develop, assure, maintain, operate, and manage software in support of Agency programs.[3] Software development projects like the SCCS seek to achieve these objectives within cost and schedule constraints.

---

[1] NASA Office of Inspector General (OIG), "NASA's Launch Support and Infrastructure Modernization: Assessment of the Ground Systems Needed to Launch SLS and Orion" (IG-15-012, March 18, 2015).

[2] NASA Aerospace Safety Advisory Panel, "Annual Report for 2014," January 28, 2015.

[3] NASA Procedural Requirements (NPR) 7120.5E, "NASA Space Flight Program and Project Management Requirements w/Changes 1- 13," August 14, 2012.

NASA's Software Engineering Process

NASA's information technology project management process is based on life cycles with key decision points at strategic phases, evolving products, decision authority, governance structure, and independent assessments. A life cycle model serves as the conceptual framework describing activities at various stages throughout a software development project from inception to retirement. Life cycle planning considers the software system as a whole and identifies the planning activities required to ensure a coordinated, well-engineered process.

At conception, a project's software needs are analyzed, including acquisition, supply, development, operation, maintenance, retirement, and supporting activities and processes. The software effort is scoped and processes, measurements, and activities are documented in accompanying software plans.

Although NASA policy does not impose a particular life cycle model on software development projects, the policy does endorse a standard set of six sequential phases that allow for gradual development and maturity over time: (1) requirements, (2) architecture, (3) design, (4) implementation, (5) testing, and (6) operations, maintenance, and retirement.[4]

Requirements

A requirement is a condition or capability the user needs to solve a problem or achieve an objective. Because it provides the basis for planning, cost estimating, and monitoring software development, defining requirements is one of the most critical phases of a software engineering project.

Architecture

Architecture is the fundamental organization of a software system, the relationship of the system's components to one another and the environment, and the principles governing the system's overall structure and evolution. A software system's architecture underpins its design and code and represents the earliest design decisions, which can be difficult and costly to change later in the development cycle. The objective of this phase is to define and formalize software dependencies within the integrated system. The quality and longevity of a software-reliant system is largely determined by its architecture.

Design

Software design focuses on creating a strong overall structure by defining the components, modules, interfaces, and data necessary for a software system to meet requirements.

Implementation

Implementation consists of employing the requirements and design into code, data, and records. At the end of this phase, all required software products should be ready for delivery, subject to modification during integration and testing.

[4] NPR 7150.2B," NASA Software Engineering Requirements," November 19, 2014.

Testing

During testing, software products are integrated into the completed software system, problems or defects (nonconformance) are identified and corrected, and the software system demonstrates that it meets requirements. Software verification and validation activities occur throughout the software life cycle and may include formal and informal reviews, peer reviews, inspections, testing, demonstrations, and analyses.

Operations, Maintenance, and Retirement

During the operations, maintenance, and retirement phase, the software system is used to achieve the objectives for which it was acquired or developed. Corrections and modifications are made to sustain operational capabilities and upgrade capacity to support users. Changes may range in scope from simple corrective actions to major modifications that require a full life cycle process. Maintenance activities sustain the software system after delivery to the customer through retirement. Planning for operations, maintenance, and retirement typically occurs throughout the software system's life cycle.

Capability Maturity Model Integration Certification

For more than 2 decades, thousands of organizations have used an organized collection of best practices and proven processes – known as Capability Maturity Model Integration (CMMI) – to guide software development efforts. Administered and marketed by Carnegie Mellon University, CMMI provides guidance on topics such as eliciting and managing requirements, decision making, measuring performance, planning work, and addressing risks. A team of trained professionals evaluates an organization's software development processes and determines whether various process areas, such as requirements management, configuration, quality, and defect prevention, have been satisfied and a CMMI rating should be awarded. Development efforts that meet CMMI standards receive a satisfactory rating and are certified for a period of 3 years. NASA requires the use of CMMI as a means of ensuring software development projects produce reliable products within cost and schedule estimates. Compliance and accreditation at CMMI Level 3 is a requirement for companies bidding on NASA software contracts.[5]

Exploration Mission-1

NASA intends to use the SCCS software on Exploration Mission-1 – the first launch of the combined SLS-Orion system scheduled for November 2018. The 22-day mission will launch without a crew from Kennedy's Launch Pad 39B to test system readiness for future crewed operations. NASA plans the first crewed flight of the combined system, known as Exploration Mission-2, no later than 2023. Orion flew its first test flight in December 2014, launching without a crew from Cape Canaveral Air Force Station on a United Launch Alliance Delta IV rocket and successfully completing a 4-hour, two-orbit trip around Earth. The test flight used command and control software developed by The Boeing Corporation and used by United Launch Alliance for the Delta rocket.

---

[5] The CMMI model identifies five levels of process maturity: initial, repeatable, defined, managed, and optimizing. A Level 3 maturity indicates a software development process characterized by requirements development, technical solution, product integration, verification, validation, process focus, process definition, training, integrated project management, risk management, and decision analysis and resolution.

Past NASA Software Development Efforts

The SCCS is not the first time NASA has attempted a software development effort of this scale. Between 1995 and 2002, the Agency spent more than $500 million on two separate efforts to update the Space Shuttle's Checkout and Control System, which was part of Kennedy's Launch Processing System that was first installed in the mid-1970s.[6] As a result of repeated cost, schedule, and performance issues, both efforts failed to meet their objectives and were substantially scaled back or cancelled prior to completion.

Core Electronics System

NASA's first attempt to incorporate the advantages of modern, state-of-the-art real-time computers, displays, software, and communications into Space Shuttle operations began in 1989.[7] Implemented by a joint NASA, Harris Corporation team, one of the objectives of the Core Electronics System was to develop a replacement Checkout, Control, and Monitor Subsystem for the Space Shuttle Launch Processing System. In February 1994, NASA announced a "realignment" of the contract with the Harris Corporation due to reductions in the Space Shuttle and International Space Station program budgets. The contract was reduced from its original estimated value of $355 million to approximately $260 million.

Checkout and Launch Control System

The second modernization attempt began in 1996 when NASA initiated a project to replace Kennedy's entire Launch Processing System. The Checkout and Launch Control System Project incorporated the work accomplished under the Core Electronics System contract and was intended to feature several major improvements over the existing Launch Processing System, including the capability to monitor more than one Orbiter from a single firing room. NASA planned to use commercial off-the-shelf (COTS) hardware and software to the fullest extent possible and estimated the new system would reduce operating costs by 50 percent while making it easier to upgrade the system in the future.[8]

With a total estimated cost of $206 million, the Project was planned as a 5-year effort with capability delivered in increments and the system fully operational by September 2001. However, the Project quickly fell behind schedule and began to experience cost overruns. In August 2002, an internal NASA assessment team concluded that rather than being less expensive than the existing system, the new system could cost $15 million more per year to operate. The assessment team also estimated the price of finishing the system could rise to $533 million and would take an additional 4 years (until 2005) to complete. With spiraling costs and an uncertain timeframe for completion, NASA canceled the Checkout and Launch Control System Project in September 2002 after investing $273 million. The Space Shuttle's original Launch Processing System – first used in April 1981 – remained in use through its final flight in 2011.

---

6 The command and control system used for the Space Shuttle was formally named the Checkout, Control, and Monitor System, but was commonly known as the Checkout and Control System.

7 Real-time computer systems are designed as reactive systems that observe changes in the environment, compute appropriate actions, and convey actions to various components so that the system operates correctly within the designated time constraint.

8 COTS is a Federal Acquisition Regulations term for items available in the commercial marketplace.

Constellation

The roots of the current SCCS development effort lie with NASA's Constellation Program, which began in 2005 and was cancelled in 2010. NASA established the Constellation Program in response to the NASA Authorization Act of 2005, which called for development of a crew capsule, heavy launch vehicle, and crew exploration vehicle to return to the Moon and serve as a stepping-stone to future exploration of Mars and other destinations.[9] The Constellation Program was intended to replace the Space Shuttles with capsule-based vehicles designed to launch on two new rockets – the Ares I and a larger, heavy-lift rocket called Ares V, which would be capable of launching lunar landers and rocket stages for Moon-bound missions.

Development of the precursor to the SCCS began in June 2005 when NASA formed a team to conduct a feasibility study for developing a new launch site command and control system for the Constellation system. Ultimately, the team selected a Standards Based Architecture approach for the development effort. This approach involved acquiring COTS components and writing the software code necessary to "glue" them together so they would communicate and interact with one another using "glue-ware." The team estimated development costs for this effort at $128 million and full life cycle costs at $326 million.

21st Century Space Launch Complex

Following cancellation of the Constellation Program, the Agency established a program office at Kennedy with the goal of converting the Center into a "21st Century Space Launch Complex." Work on the launch command and control system was approved to continue as part of the GSDO Program, but with a limited scope that focused on ground support equipment capabilities at Kennedy's Launch Complex 39. The intent was to deliver a basic command and control capability suitable for both Agency and commercial launches. After NASA announced the SLS and Orion programs in 2010, the GSDO Program assumed responsibility for developing the launch command and control system for the programs.

[9] NASA Authorization Act of 2005, Pub. L. No. 109-155, December 30, 2005.

Cost Increases and Schedule Delays Have Adversely Affected SCCS Development Effort

The SCCS development effort has significantly exceeded initial cost estimates and is more than a year behind schedule. In addition, several planned capabilities have been deferred because of cost and timing pressures. The root of these issues largely results from how NASA implemented the Standards Based Architecture approach, which requires integration of multiple interconnected software products or, in some cases, parts of these products. "Gluing" together these disparate products has turned out to be more complex and expensive than NASA anticipated. Based on our analysis, the SCCS development effort will result in a system with reduced functionality for the first launch of the combined SLS-Orion scheduled for 2018. Moreover, the SCCS development effort is at risk of incurring further cost growth and causing potential schedule delays for other SLS-Orion launches.

SCCS Development Costing More and Taking Longer Than Expected

Compared to its fiscal year (FY) 2012 baseline, development costs for the SCCS have increased approximately 77 percent and the schedule has slipped 14 months. NASA develops its budget authority as part of the annual Planning, Programming, Budgeting, and Execution (PPBE) process. As shown in Table 1, the estimated cost of developing and operating the SCCS Launch Control System increased from $117.3 million in FY 2012 (PPBE 13) to $207.4 million in FY 2015 (PPBE 17).[10] According to NASA's most recent budget estimates, the Agency will spend almost three times the planned amount on the SCCS in FY 2016 and more than four times the following year.

Table 1: Launch Control System Funding

Budget Authority	Fiscal Year (dollars in millions)													
	2012[a]	2013[a]	2014[a]	2015[a]	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	Total
PPBE 13	$17.5	$12.8	$11.9	$11.8	$10.8	$4.9	$5.4	$7.1	$5.5	$8.4	$3.9	$5.2	$5.1	$7.0	**$117.3**
PPBE 17	$15.0	$26.2	$24.0	$27.0	$30.6	$23.6	$15.3	$11.5	$8.1	$5.9	$4.8	$4.9	$4.9	$5.6	**$207.4**

Source: NASA budget data.

[a] Actual FY dollars.

[10] The SCCS Launch Control System is the software that provides the basic functionality for a launch command and control system. During operations, the system provides consoles, connectivity, the ability to command end items, process measurements and telemetry, record and retrieve data, and reports on the overall status of the SCCS.

In an effort to avoid further cost increases and schedule delays, SCCS's Project managers have deferred several planned capabilities, including some system summary status displays and fault tolerance levels. Summary status displays allow monitoring of the operational configuration and measurements of the rocket, spacecraft, and ground systems, while fault tolerance levels provide engineers with a technical explanation of the failure and possible corrective action to avoid an unsafe situation.

Release of a fully operational version of the SCCS software has been delayed from July 2016 until September 2017 – a 14-month schedule slip. Table 2 presents a high-level summary of the schedule, by function, of the multiple software versions required to achieve full operational status for the SCCS. The various release dates reflect when the GSDO Program will have validated that the software performs as intended in an operational environment.

Table 2: SCCS Software Release Schedule

Version	Function	Fiscal Year				
		2012	2013	2014	2015	2016
1	Basic Command and Control	November 2011	May 2013	July 2013[a]		
2	Non-Hazardous Command and Control			August 2014	September 2014[a]	
3[b]	Hazardous Command and Control			July 2015	August 2015	June 2016
4[c]	Operational Software			July 2016	October 2016	September 2017

Source: NASA OIG analysis of NASA schedule data.

[a] Actual date.

[b] Denotes software versions 3.0–3.4.

[c] Denotes software versions 4.0–4.1.

As is typical of software development efforts, the SCCS is being developed iteratively beginning with version 1.0 and progressing to versions 2.0, 3.0, and 4.0. Each software version also contains sub-releases or "builds" – designated as, for example, version 3.4 – that contain additional content. Each iteration builds on previous versions and will be integrated into the overall system that will operate and monitor equipment managed by the GSDO, SLS, and Orion programs. SCCS 4.0 will be the culmination of the software development process for Exploration Mission-1 and the version NASA intends to use for that mission. As of the end of FY 2015, the SCCS's Project team had completed development of versions 1.0 and 2.0 and was working on version 3.0.

SCCS software version 3.0 will represent four builds that, when completed, will contain all the requirements needed to support hazardous testing and operations of ground support equipment at Kennedy as well as the SLS gateway needed for testing at the Marshall Space Flight Center's (Marshall) Software Integration Lab.[11] Each of version 3.0's four builds is tied to a specific customer need:

[11] The SLS gateway processes raw telemetry information into usable data for flight controllers. The hazardous testing and operations subsystem and SLS gateway use the SCCS command and control infrastructure to provide the human-machine interface for remote and local control and to monitor ground support equipment subsystems such as liquid hydrogen, liquid oxygen, ground cooling systems, and environmental control.

- SCCS 3.1 will be used by ground support personnel to perform remote testing at the Multi-Payload Processing Facility, which fuels the Orion spacecraft and services spacecraft upon their return to Earth.
- SCCS 3.2 will support a single hazardous testing and operations subsystem at the Multi-Payload Processing Facility.
- SCCS 3.3 will support multiple hazardous testing and operations subsystems at the Multi-Payload Processing Facility and provide the SLS gateway needed for development and testing of the ground and flight application software at the Marshall Software Integration Lab.
- SCCS 3.4 will have the capability of filling the fuel storage tanks (cryo-spheres) used for launch at Launch Pad 39B.

Version 4.0 will include two builds (4.0 and 4.1), which, when completed, will contain the final pieces of the SCCS needed for vehicle processing and launch operations. This version will also deliver the remaining vehicle gateways, integration of third party electrical ground support equipment, and hardware and software needed to perform testing at the Integrated Test Lab in Denver, Colorado.[12] Version 4.0 will contain other software enhancements such as flight instrumentation processing and launch countdown.

The GSDO Program's Operational Readiness Review is scheduled for April 2018, only 5 months before the Program's internal working milestone for Exploration Mission-1. During this review, NASA will evaluate whether the flight and associated ground systems are ready for a safe, successful launch. Federal and NASA guidance require information systems such as the SCCS to have an Authorization to Operate (ATO) in accordance with National Institute of Standards and Technology (NIST) guidance.[13] Security certification and accreditation are important elements of NASA's risk management process and an integral part of the Agency's information security program. Following NIST guidance, NASA plans to complete a security certification, conduct an accreditation review, and obtain an ATO to help ensure the SCCS software will be operated and monitored appropriately. Although it has made progress on information technology security processes, the GSDO Program will require waivers for items such as the shared/group account logon and two-factor authentication to obtain the ATO.[14]

Software Nonconformances

As with any large scale, complex software development project, schedule and costs are impacted by the number of nonconformances and remaining development tasks. As of the end of FY 2015, the SCCS Project team had spent 11,360 hours addressing nonconformances with an estimated 12,570 additional hours required to resolve pending issues.[15] In addition to addressing

---

[12] Lockheed Martin's Integrated Test Lab is used to demonstrate the successful integration of avionics hardware, flight software, simulation, and ground support equipment.

[13] The ATO is a formal declaration by a Designated Approving Authority that authorizes operation and explicitly accepts the risk to Agency operations. The ATO is signed after a Certification Agent certifies that the system has passed all requirements to become operational. Failure to receive an ATO indicates major weaknesses or deficiencies in the security controls employed in the information system. The ATO should be in accordance with the following guidance: NIST, "Guide for Applying the Risk Management Framework to Federal Information Systems" (Special Publication 800-37 Revision 1, February 2010).

[14] In a shared/group logon, multiple people use a single logon identity. Two-factor authentication requires the user to have two types of credentials, such as a password and token or fingerprint, before accessing the system.

[15] These figures do not include work that may be required to address nonconformances that may occur during future software builds. The GSDO Program estimates it takes roughly 30 hours to remedy a single nonconformance.

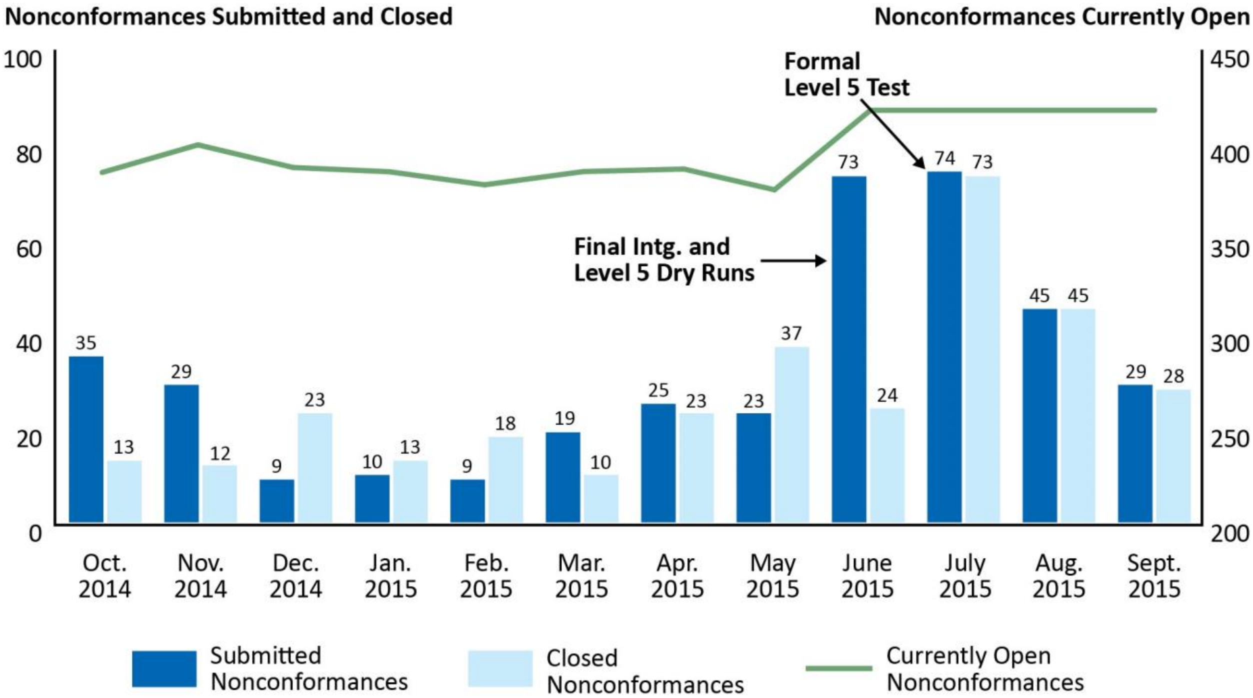
nonconformances, NASA officials estimate that it will take programmers an additional 49,176 hours to complete development of version 4.0. Correcting nonconformances diverts resources from development tasks, which can affect project schedules. While NASA is confident it can address both nonconformances and new development tasks in a timely manner, we found the assumptions underlying the Agency's schedule estimates optimistic.

Specifically, because of the number of nonconformances that need to be addressed and the development work that remains to be done, we have doubts whether the GSDO Program will be able to produce a fully operational software package by the integrated test date of October 2017, when the Program is scheduled to verify and validate that the information systems used in the Vehicle Assembly Building, the Mobile Launcher, and Launch Pad 39B are operational and functioning properly. Missing this date may mean deferral of additional software functionality. Although all software nonconformances are assessed, the GSDO Program does not have a method to forecast the date by which significant nonconformances must be corrected to avoid impacting the Program's schedule.

Historically, GSDO's Program managers have allocated 70 percent of developer time to new development and 20 percent to addressing nonconformances, with the remaining 10 percent spent on other activities. Although managers could allocate more programming time to nonconformances, this would reduce the time spent on new software development efforts, which in turn would impact the overall schedule. The GSDO Program is attempting to mitigate the schedule threat by analyzing nonconformances to determine their root causes and improving its ability to predict new nonconformances and build time into the schedule to address them. The Program hopes this strategy will detect and enable engineers to resolve discrepancies earlier in the development cycle.

As shown in Figure 1, the number of new and resolved nonconformances were virtually equivalent between July and September 2015. In other words, new nonconformances were being discovered as fast as the team could address previous ones. However, we found that since September 2015, when the team completed integration testing on build 3.0, the backlog of open nonconformances has stabilized, with fewer nonconformances being identified. Although this indicates a general improvement in software performance for build 3.0, several additional iterations of the software remain to be developed and it is not clear whether the previous pattern of a higher rate of nonconformances will recur.

Figure 1: SCCS Nonconformances



Source: NASA OIG presentation of GSDO Program information.

Note: Level 5 is the testing and integration activities the development team performs after each SCCS delivery.

Reduced Software Functionality

Since the beginning of SCCS development, and as recently as February 2016, the GSDO Program has repeatedly reprioritized, replanned, reduced, or eliminated capabilities from the software in a continuing effort to balance technical capabilities against schedule and cost issues. Early in development, many critical capabilities such as the ability to provide ground systems data to mission systems and to detect and isolate fault notifications were eliminated. However, the GSDO Operations Directorate raised concerns that too much capability had been reduced and that the risk posture for launching Exploration Mission-1 was no longer acceptable. Accordingly, the GSDO Program initiated two efforts to recapture some of the reduced capability.

First, in 2014, the GSDO Program implemented a concept known as “buy-back,” which provided the SCCS with an additional $3.7 million of funding in FYs 2014 and 2015 to restore the highest priority items. Second, GSDO officials established a process to identify and classify development tasks by priority and time required for completion. Specifically, the GSDO Program formed a Delivery Discrepancy Panel that classified software tasks as essential – required to get to the next step in development – or desirable – does not impact deliverables. Using this criteria, the Program developed prioritization levels from 1 to 200, with level 1 tasks considered highest priority, to organize the several thousand outstanding development tasks. GSDO Program officials told us that initially they believed the Program would be able to complete tasks assigned a prioritization level of 180 or less in time for Exploration Mssion-1. However, as development progressed, they have scaled back this estimate to tasks with a priority level of less than 140.

As a result of these efforts, the GSDO Program was able to reinstate several critical capabilities that had either been eliminated or deferred to Exploration Mission-2, including the stale data indicator and functionality that will help performance analysis, fault isolation, and quality assurance.[16] Without the stale data indicator, console operators would have had to rely on alternate methods to determine if vehicle or ground measurement samples were current and usable. Depending on when the stale data was detected in the operations cycle, it may have forced emergency "safing" or a launch countdown scrub.[17] In addition, during the course of our audit, GSDO Program managers decided to buy-back the ability to send real-time information to Marshall on cryogenic fuel levels in the SLS's storage tanks because they determined data analysis efforts might be hampered by manual transmission of this data.

Despite the buy-back and work prioritization efforts, the software NASA plans to use for Exploration Mission-1 will not have all the capabilities the Agency originally planned. For example, the GSDO Program deferred certain data protection capabilities designed to limit access to certain kinds of astronaut health and other sensitive data. The Program also deferred enhancements aimed at reducing the amount of time and effort needed to develop software. In addition, the ability to automatically detect the root cause of specific failures has been deferred. Without this information, it will be more difficult for controllers and engineers to understand and quickly diagnose and resolve issues. While these capabilities may not be critical for Exploration Mission-1, deferment reduces the opportunity to prove the capabilities prior to crewed launches. Moreover, although NASA officials believe the SCCS will operate safely without these capabilities, they acknowledge the reduced capability could affect the ability to react to unexpected issues during launch operations and potentially impact the launch schedule.

While the GSDO Program has been able to reinstate many deferred capabilities, the process continues to be fluid, and the extent to which the Program will have to eliminate or defer specific capabilities beyond Exploration Mission-1 is uncertain. Even with the additional funding, the SCCS is showing a schedule risk to version 4.0. As of the end of FY 2015, version 4.0 remained 3,320 hours "out of the budget box" – meaning there is more estimated work than time and staff available to perform it under the current timetable and funding level. If this situation does not change, the GSDO Program may have to further reduce content and functionality.

Complexity of Architecture Implementation Hampers Software Integration Efforts

We found the primary reason for the schedule delays, cost increases, and performance shortfalls is the manner in which project managers implemented the SCCS's software architecture. Specifically, integration of multiple software products has proven substantially more complex than anticipated.

[16] Quality assurance is a planned and systematic pattern of all actions necessary to provide adequate confidence that an item conforms to established technical requirements.

[17] Safing involves taking a number of immediate actions to prevent damage to or complete loss of the spacecraft.

Decision to Develop Software Architecture Based on Trade Studies

Over the last 10 years, NASA conducted a series of trade studies to determine the most suitable architecture for the SCCS. The purpose of these studies was to analyze alternate solutions that could reduce licensing, development, and maintenance costs while still meeting requirements. The Agency also used these studies to determine whether developing the software in-house, buying a COTS product, or some combination of the first two options would make the most sense. NASA established the following strategic objectives for the trade studies:

- Effectively transition from the Space Shuttle Program to deep space exploration operations that will enable NASA missions to extend human existence beyond the Moon, to an asteroid, to Mars, and across the solar system.
- Provide an evolutionary path to meet the projected 40-year life expectancy of the Constellation Program.
- Minimize cost of ownership to the Constellation Program.

The "make-or-buy" decision is a strategic selection between developing software in-house or purchasing from an outside supplier. Factors such as uniqueness, desire to maintain quality control, workforce proficiency, and need to protect proprietary technology favor an in-house build. On the other hand, because of the reduced need for coding, buying software from an outside supplier is often less costly and time-consuming and may enable users to take greater advantage of new technologies. In addition, operational and cost factors such as licensing and sustaining maintenance must be considered regardless of whether the software is developed in-house or purchased commercially.

As discussed, NASA decided to develop the SCCS software by purchasing multiple COTS software packages and "gluing" them together with code written by NASA civil servants and contractors. SCCS Project managers told us this decision was motivated by several factors. First, managers did not want to rely on a single company to provide the software because if that company encountered financial difficulties or stopped providing technical support NASA's space exploration efforts could be negatively impacted. In addition, at the time the architecture decision was made in June 2006, SCCS Project managers believed the effort to integrate the various COTS products would not be overly time-consuming or technically complex.

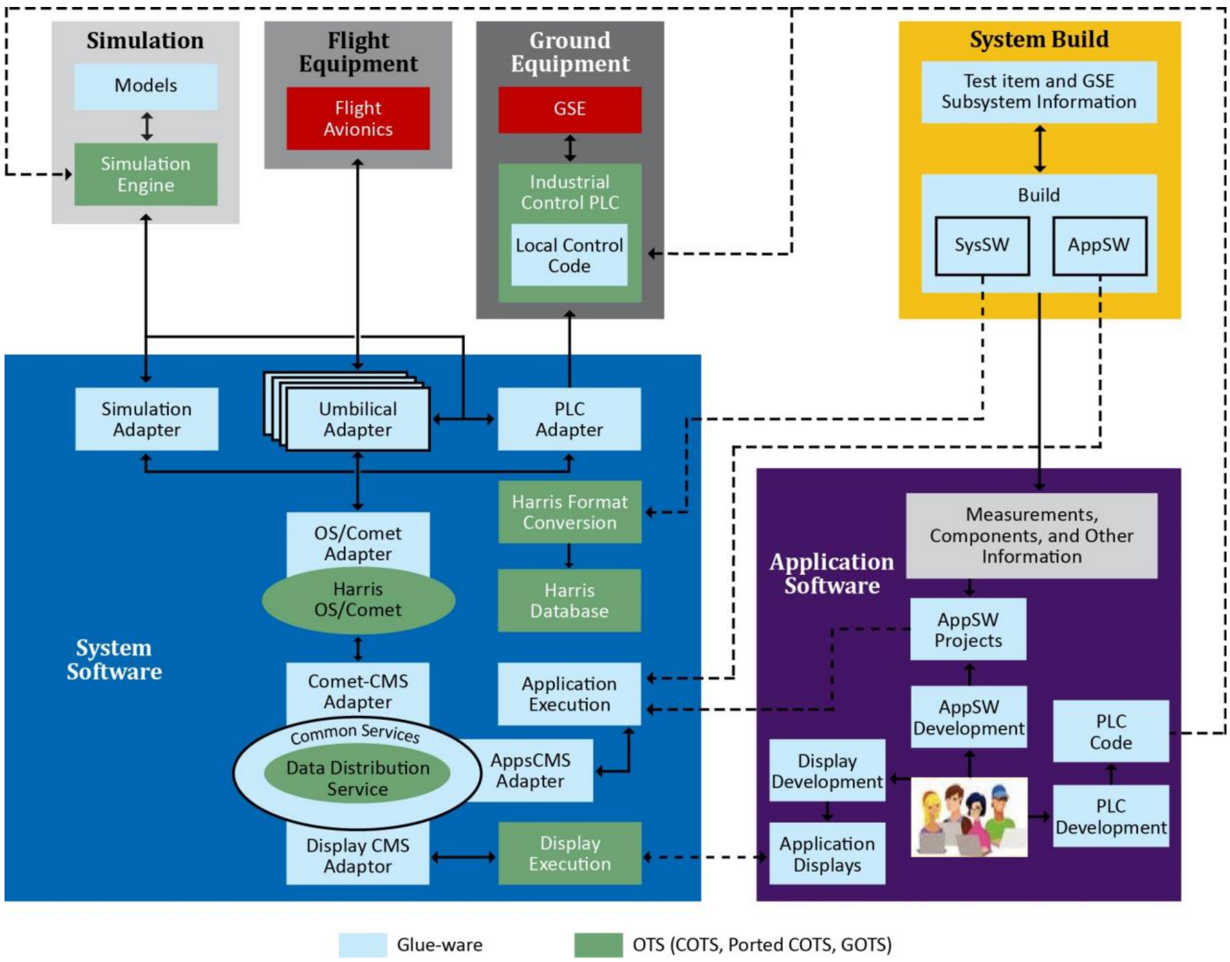
Integration Problems with Software Architecture

While the architecture decision made in 2006 may have been reasonable based on what NASA managers knew at the time, it is now clear that the Agency underestimated the complexity of the software integration activities the approach would require and, as a result, the SCCS development effort has encountered significant technical challenges. Specifically, the GSDO Program underestimated the amount and complexity of the "glue-ware" needed to integrate the nine major software packages that make up the SCCS. Generally, the need to glue together multiple complex software packages can increase the cost and lengthen the schedule of a software development effort. Moreover, utilizing "glue-ware" when a significant amount of customization is required has been shown to increase the likelihood of defects resulting in lower system quality.

Instead of "gluing" together fully integrated COTS products, the GSDO Program tried to take the best features from multiple software products and glue them together by writing code in-house – akin to taking automobile parts not designed to work together from several different brands and attempting to assemble a new car. Developing the SCCS software in this manner has cost millions more than expected, resulted in a schedule that continues to slip, and will produce a product with less functionality than planned.

Figure 2 depicts the general software structure of the SCCS and the extensive amount of "glue-ware" required to integrate the various software components indicated by the light blue boxes. SCCS Project managers told us that the amount and complexity of the "glue-ware" required to make the software operate efficiently was significantly more than expected. Specifically, they said that as of January 2016 they had developed 2.5 million lines of "glue-ware" code for the nine products they are attempting to integrate with almost two more years of development activity planned. In comparison, NASA reengineered the Hubble Space Telescope command and control system by integrating 30 products with approximately 500,000 lines of "glue-ware" code.

Figure 2: SCCS Software Structure



Source: NASA OIG presentation of GSDO Program information.

Note: Figure acronyms include the following: Application Content Management System (AppsCMS), Application Software (AppSW), Content Management System (CMS), Commercial Off-the-Shelf software (COTS), Government Off-the-Shelf software (GOTS), Ground Support Equipment (GSE), Off-the-Shelf software (OFS), Programmable Logic Controller (PLC), and System Software (SysSW).

Several industry studies have reported on the complexities of integrating multiple commercial systems.[18] For example, these studies found that

- significantly customizing commercially available software products introduces risk and complexity and can negate the benefit of using a commercial product;
- bug fixing can be particularly problematic when integrating commercial products especially when multiple vendors are involved or when modifications have been made to the commercial components;
- the major cost, effort, and schedule drivers for "glue-ware" development is not unlike the drivers associated with custom development, but the productivity of the development team needs to be adjusted to account for unfamiliarity with the commercial systems being used and the requirement for vendor support and cooperation in solving integration problems; and
- as the number of commercial components and vendors' increases, the complexity and thus the effort and schedule of integration increases.

Two of the key software packages the SCCS Project team is "gluing" together are Harris Corporation's OS/Comet and Real-Time Innovations, Inc.'s Connext Data Distribution Service. The OS/Comet software package provides the processing capabilities for sending commands to the flight vehicle and associated hardware, while the Connext Data Distribution Service is the network programming middleware that enables computers on the network to communicate with one another. According to GSDO officials, the SCCS Project chose to integrate these packages because none of the commercial systems then available could be used "as is, out of the box" due to the unique, NASA-developed propriety network communication protocol known as Command, Control, Communication, and Information.[19] This protocol is used by the Orion spacecraft and needed for ground communications.

The extent of the integration issues the SCCS Project team has encountered are reflected in the GSDO Program's inability to achieve a CMMI rating for the software until June 2015 – 5 years after the development effort began. Although a CMMI rating is required by NASA at the earliest stages of the software development process, the SCCS was unable to achieve the rating for 5 years because the Project could not show a satisfactory rate of maturity. In rejecting earlier requests to certify the software development process, CMMI officials cited such concerns as the SCCS's inability to measure code against defects, the lack of a documented process for software builds, and the absence of a realistic plan to assess the allocation of resources. As a result, the SCCS did not have the benefit of CMMI-based process improvements, such as schedule and budget predictability and reduction of software defects and bugs, during crucial periods of the development process.

---

[18] Minkiewicz, Arlene "6 Steps to a Successful COTS Implementation," PRICE Systems, available at http://www.compaid.com/caiinternet/ezine/cots-am.pdf (accessed February 24, 2016), and Couts, C. Todd and Gerdes, Patrick F., "Integrating COTS Software: Lessons from a Large Healthcare Organization," available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5439513 (accessed February 24, 2016).

[19] The interface defines the transmission of telemetry data files to and from spacecraft, its data storage, file management services, and various other capabilities for operating in a wide variety of mission configurations. The Command, Control, Communication, and Information protocol is a propriety network communication protocol developed at, and unique to, Kennedy. The protocol encompasses the technologies of communications, distributed information systems, command and control, decision support, and information operations.

Available COTS Software May Provide More Efficient Solution

Experience has shown that the quality and longevity of a software-reliant system is largely determined by the architecture of its software. NASA made its decision regarding the architecture of the SCCS software nearly a decade ago and has continued on that path. In our view, this may no longer be the most prudent course of action given the significant advances in commercial command and control software over the last 10 years. Specifically, command and control software technology has matured to the point where COTS products may provide much of the functionality needed to launch the SLS and Orion with relatively little modification. Indeed, the two companies under contract with NASA to deliver supplies to the International Space Station – Orbital Sciences Corporation and Space Exploration Technologies – both use COTS products to accomplish their missions.

Early in our audit, we questioned GSDO Program officials regarding the viability of procuring an integrated, commercially available command and control software suite in lieu of continuing development of the SCCS. Program officials said they believed there was insufficient time to make such a significant change without impacting the Exploration Mission-1 launch schedule; however, additional schedule slips and delayed delivery of software capabilities have caused the GSDO Program to reconsider. For example, the Program has decided to utilize software developed by United Launch Alliance to interface with the SLS's cryogenic propulsion avionics. Additionally, the Program is studying whether utilizing part of the Lockheed Martin command and control system used for the December 2014 Orion test flight would reduce risk to Exploration Misson-1.

Delays in the SLS and Orion programs may provide the GSDO Program with additional time to reevaluate its software development strategy. For example, NASA has announced that although it is still working toward a 2021 launch date for the Exploration Mission-2, its actual commitment date for the launch is 2023.

Despite cost increases, schedule delays, reduced capability, and advancements in COTS command and control software, GSDO officials have continued to develop the SCCS using the approach NASA selected nearly 10 years ago. In our judgment, the GSDO Program's reluctance to change course reflects a cultural legacy at NASA of over-optimism and over-promising what the Agency can achieve in a specific timeframe. As we noted in an August 2012 report, while optimism and a can-do attitude is essential to producing the types of unique space flight projects NASA undertakes, it can also lead managers to underestimate the amount of time and money needed to overcome significant technical challenges.[20]

[20] NASA OIG, "NASA's Challenges to Meeting Cost, Schedule, and Performance Goals" (IG-12-021, September 27, 2012).

CONCLUSION

The SCCS software under development by the GSDO Program is intended to provide end-to-end command and control infrastructure for processing and launching the next generation of NASA space flight, including Exploration Mission-1 scheduled for November 2018. In the past, NASA has encountered difficulties with large and complex command and control software development efforts, failing on two occasions to meet expected requirements despite spending more than $500 million. In something of a repeat of this pattern, the SCCS development effort is more than 1 year behind schedule and significantly over cost, and several planned software capabilities have been deferred. The root of these issues largely results from how NASA implemented the Standards Based Architecture approach, which requires integration of multiple interconnected software products or, in some cases, parts of these products. Unfortunately, accomplishing this has turned out to be much more complex than anticipated. In our view, NASA should consider alternative software development approaches such as buying an integrated commercial product or reducing the number of software products requiring integration through "glue-ware."

GSDO Program officials expressed concern about changing the SCCS development approach a little more than 2 years before the scheduled launch date of Exploration Mission-1. We acknowledge that altering course at this point would be ambitious. However, we are concerned that if SCCS schedule and functionality issues are not addressed soon, the software development effort may suffer a similar fate as past efforts to upgrade launch command and control software at Kennedy. Although GSDO officials told us pivoting to an integrated COTS solution at this point could potentially delay completion of the system by a year, we believe now is the time to reassess the status of the SCCS development effort and put a significant change in approach on the table for discussion. We suggest NASA consider a parallel process that would continue the minimum development efforts necessary to launch Exploration Mission-1 while at the same time initiating an independent evaluation to determine whether available commercial command and control products could replace current efforts with greater functionality at a lower life cycle cost. Such an evaluation would also inform the GSDO Program's approach for Exploration Mission-2 – the first crewed lunar orbit mission scheduled for 2023. Although the GSDO Program plans to add functionality in time for this mission, we anticipate the SCCS Project will encounter many of the same difficulties it is currently experiencing.

Recommendation, Management's Response, and Our Evaluation

In a draft version of this report, we recommended the Associate Administrator for Human Exploration and Operations commission an independent assessment to evaluate the status of the SCCS software development effort and determine the necessary steps to reduce the risk of further cost, schedule, and performance issues, including consideration of acquiring commercial command and control software to replace some or all of the system currently under development.

In their response to the draft, NASA officials agreed to conduct an independent assessment of the command and control system once the software for Exploration Mission-1 is successfully delivered. They also acknowledged they had underestimated the difficulty and complexity of integrating COTS products with custom code and noted they had initiated a series of process improvements during the course of our audit. Finally, they pointed to a 2013 review by The Aerospace Corporation (Aerospace) that confirmed that the Standards Based Architecture NASA is developing for the SCCS is generally sound.

We acknowledge Aerospace's assessment and do not take issue with the conclusion that the software NASA is developing is generally sound. However, as stated in our report, the integration of multiple, interconnected software products – or in some cases parts of products – has resulted in both cost and schedule risk to the SCCS Project. Moreover, significant advances in commercial command and control software since NASA made its architecture decision 10 years ago suggest the current path may no longer be the most prudent course of action. In addition, we note Aerospace itself recommended an annual independent analysis of the Project's cost and schedule – an analysis that has yet to be conducted.

That said, we consider management's plan to conduct an independent assessment after delivery of the software for Exploration Mission-1 responsive to our recommendation. Therefore, the recommendation is resolved and will be closed upon completion and verification of the proposed corrective action. NASA's full response is reproduced in Appendix B. Technical comments provided by the Agency have also been incorporated, as appropriate.

Major contributors to this report include, Ridge Bowman, Space Operations Director; G. Paul Johnson, Project Manager; Loretta Atkinson, Project Manager; Linda Hargrove; Jonathan Flugel; Frank Martin; and Dimitra Tsamis.

If you have questions about this report or wish to comment on the quality or usefulness of this report, contact Laurence Hawkins, Audit Operations and Quality Assurance Director, at 202-358-1543 or laurence.b.hawkins@nasa.gov.

Paul K. Martin
Inspector General

APPENDIX A: SCOPE AND METHODOLOGY

We performed this audit from April 2015 through February 2016 in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for our findings and conclusions based on our audit objectives. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objectives.

In this audit, we evaluated whether NASA is effectively managing the SCCS software development effort at Kennedy Space Center and as such, our review was conducted at Kennedy. To accomplish this review, we spoke with key GSDO Program officials and staff regarding the SCCS software development process. Additionally, we interviewed commercial companies involved with command and control software. We also reviewed various studies concerning the SCCS.

We obtained and examined applicable documents and verified compliance with the NASA Authorization Act of 2010 (Pub. L. No. 111-267, October 11, 2010), and NASA policies. The documents we examined included the following:

- "National Aeronautics and Space Administration (NASA) Authorization Act of 2010," Pub. L. No. 111-267, October 11, 2010
- NPR 7120.5E, "NASA Space Flight Program and Project Management Requirements (w/Changes 1-13)," August 14, 2012
- NPR 7150.2B, "NASA Software Engineering Requirements," November 19, 2014
- NPR 8705.2B, "Human-Rating Requirements for Space Systems (w/change 4 dated 8/21/2012)," May 6, 2008
- NASA/SP-2013-604, "NASA Software Engineering Benchmarking Study," May 2013
- GSDO-ACO-1010, Revision C, "Ground Systems Development and Operations Program, Architectures and Concept of Operations Document," October 1, 2014
- GSDO-C3R-3010, "Spaceport Command and Control System Project, Launch Control Subsystem Content Priority Lists"
- GSDO-MVVP-1042-01, "Ground Systems Development and Operations Program Master Verification and Validation Plan, Volume 1 – Program Verification and Validation Policies and Guidelines," February 13, 2013
- GSDO-MVVP-1042-02, "Ground Systems Development and Operations Program Master Verification and Validation Plan, Volume 2 – Program Verification and Validation Plan," December 10, 2013
- GSDO-MVVP-1042-05, "Ground Systems Development and Operations Program Master Verification and Validation Plan, Volume 5 – Command, Control, Communications, and Range," September 20, 2013

- GSDO-PLN-1046, Revision A, "Ground Systems Development and Operations Program Software Management Plan," September 4, 2013
- Aerospace Safety Advisory Panel, "Annual Report for 2014," January 28, 2015
- The Aerospace Corporation, "Independent Assessment of the Ground Systems Development & Operations (GSDO) Spaceport Command & Control System (SCCS)," Justin F. McNeill, Jr., September 12, 2013
- NIST Special Publication 800-37 Revision 1, "Guide for Applying the Risk Management Framework to Federal Information Systems," February 2010

Use of Computer-Processed Data

We used limited computer-processed data to perform this audit. Specifically, we reviewed various schedules, technical performance reports, and budget data from NASA's financial system. Generally, we concluded the data was valid and reliable for the purposes of this review.

Review of Internal Controls

We reviewed and evaluated internal controls, including applicable Federal laws and NASA policies and procedures. We considered the reviewed internal controls to be adequate.

Prior Coverage

During the last 5 years, the NASA OIG and the Government Accountability Office (GAO) have issued three reports and one testimony of significant relevance to the subject of this report. Unrestricted reports can be accessed at https://oig.nasa.gov/audits/reports/FY16/index.html and http://www.gao.gov, respectively.

NASA Office of Inspector General

NASA's Launch Support and Infrastructure Modernization Efforts: Assessment of the Ground Systems Needed to Launch SLS and Orion (IG-15-012, March 17, 2015)

NASA's Independent Verification and Validation Program (IG-14-024, July 16, 2014)

NASA's Challenges to Meeting Cost, Schedule, and Performance Goals (IG-12-021, September 27, 2012)

Government Accountability Office

Information Technology, Leveraging Best Practices to Help Ensure Successful Major Acquisitions (GAO-14-183T, November 13, 2013)

APPENDIX B: MANAGEMENT'S COMMENTS

National Aeronautics and Space Administration
Headquarters
Washington, DC 20546-0001

March 23, 2016

Reply to Attn of: Human Exploration and Operations Mission Directorate

TO: Assistant Inspector General for Audits

FROM: Associate Administrator for Human Exploration and Operations

SUBJECT: Agency Response to OIG Draft Report "Audit of the Spaceport Command and Control System" (A-15-008-00)

NASA appreciates the opportunity to review and comment on the Office of Inspector General (OIG) draft report entitled "Audit of the Spaceport Command and Control System" (A-15-008-00), dated February 26, 2016.

In the draft report, the OIG makes one recommendation addressed to the Associate Administrator for Human Exploration and Operations Mission Directorate (AA/HEOMD). Specifically, the OIG recommends the AA/HEOMD:

Recommendation 1: Commission an independent assessment to evaluate the status of the Spaceport Command and Control System (SCCS) software development effort and determine the necessary steps to reduce the risk of further cost, schedule, and performance issues, including consideration of acquiring commercial command and control software to replace some or all of the system currently under development.

Management's Response: Concur. Since the audit was conducted, NASA has instigated a series of process improvements that are bearing positive results. Most recently, NASA achieved successful early completion of SCCS 3.2 validation. This important release contains all of the capability required for non-hazardous ground support equipment testing to be performed. The NASA SCCS 3.3 team also recently made an important on-time software delivery that enables application software developers to begin coding procedures and displays that will be used to test and checkout Space Launch System (SLS) flight hardware. Additionally, SCCS 3.4 development and test is on track to complete in June of this year. This sustaining release provides an upgraded record and retrieval capability for firing room operators and enables the team to perform integrated hazardous operations for ground support equipment. SCCS 4.0 represents the last of the major SCCS software deliveries – it will provide the interface to process commands and data for the Orion spacecraft and monitor data from the launch vehicle's Interim Cryogenic Propellant Stage. Coding for SCCS 4.0 is on schedule to complete in the summer of 2016 with final integration and test expected to wrap up in January of 2017.

Consistent with the audit findings, NASA acknowledges the challenge of developing large-scale, real-time command and control system software for launch vehicle and spacecraft processing. NASA also agrees that the difficulty and complexity of integrating commercial off-the-shelf software (COTS) products with custom code was more challenging than estimated. At the same time, the flexibility associated with a standards-based architecture has proven beneficial when the project needed to replace COTS products due to obsolescence, performance concerns, and/or vendor support concerns. Furthermore, as Ground Systems Development and Operations (GSDO) is nearing the completion of coding for the last major release, NASA does not believe that it makes sense to re-open the architecture strategy for Exploration Mission (EM)-1. As was noted during discussions with the OIG team but not reflected in the draft report, NASA commissioned an independent team of industry experts from the Aerospace Corporation to assess the SCCS architecture in 2013. That independent assessment confirmed that the standards-based software architecture being implemented in SCCS is generally sound.

NASA welcomes an independent assessment from a team of real-time command and control system experts. It is important that NASA continually evaluates the architecture and leverages COTS software applications as they evolve. NASA also wants to take full advantage of custom software that has been developed across all Agency programs. Therefore, NASA intends to conduct an independent assessment of command and control systems once the software for EM-1 is successfully delivered, focused on opportunities that can be implemented in time for the EM-2 flight of the SLS/Orion.

Estimated Completion Date: September 30, 2018

NASA has reviewed the draft report for information that should not be publicly released. As a result of this review, we have not identified any information that should not be publicly released. Technical comments in response to the draft report have been provided separately.

Once again, thank you for the opportunity to review and comment on the subject draft report. If you have any questions or require additional information regarding this response, please contact Michelle Bascoe on (202) 358-1574.

William H. Gerstenmaier

cc:
Director, Kennedy Space Center/Mr. Cabana
Ground Systems Development and Operation Program/Mr. Bolger

APPENDIX C: REPORT DISTRIBUTION

National Aeronautics and Space Administration

Administrator
Deputy Administrator
Associate Administrator
Chief of Staff
Associate Administrator, Human Exploration and Operations Mission Directorate
Director, Kennedy Space Center
    Manager, Ground Systems Development and Operations Program

Non-NASA Organizations and Individuals

Office of Management and Budget
    Deputy Associate Director, Energy and Space Programs Division

Government Accountability Office
    Director, Office of Acquisition and Sourcing Management

Congressional Committees and Subcommittees, Chairman and Ranking Member

Senate Committee on Appropriations
    Subcommittee on Commerce, Justice, Science, and Related Agencies

Senate Committee on Commerce, Science, and Transportation
    Subcommittee on Space, Science, and Competitiveness

Senate Committee on Homeland Security and Governmental Affairs

House Committee on Appropriations
    Subcommittee on Commerce, Justice, Science, and Related Agencies

House Committee on Oversight and Government Reform
    Subcommittee on Government Operations

House Committee on Science, Space, and Technology
    Subcommittee on Oversight
    Subcommittee on Space

(Assignment No. A-15-008-00)