Collaborative Software Development and Sustaining Engineering: An Improved Method to Meet the NASA Mission.

David Mann, Ph.D.
United Space Alliance (USA)
Launch Processing System Engineering
8550 Astronaut Boulevard, MS: USK-489
Cape Canaveral, FL 32920-4304

ABSTRACT

This paper reports on the Space Shuttle, Record and Playback Subsystem (RPS) upgrade project turnaround brought about through extensive collaborative software development. The new project and systems engineering methodologies employed on this project resulted in many positive effects over the status quo method employed to develop and upgrade systems. These effects include; 1) a reduction in the initial software development costs, 2) a reduction in the development timeline, 3) improved marketability of the software technology developed, 4) improved product quality deployed to operations, and 5) improved maintainability. Attributes within each of the aforementioned are examined in support of these assertions.

Prior to implementing this new method, the RPS upgrade project had been under development for seven years using the standard software development method. This involves developing custom applications using Commercial Off The Shelf (COTS) hardware, operating systems and compilers. A change in strategy was effected on this pathfinder project by adopting a COTS telemetry ground station software package to provide basic ground station functionality and building additional required capabilities to complete the project. The merits of having employed this methodology are explored using the probable outcome of continued custom software development as a basis for comparison .

This collaboration between the United Space Alliance (USA) and AP Data Systems Inc.(an AP Labs company), resulted in software innovations in FM and PCM processing software as well as general ground station management software. The four technology transfer submittals for new software innovations resulting from this collaboration are discussed.

KEYWORDS

Telemetry, Software Development, Collaboration, Project Management or Technology Transfer.

INTRODUCTION

Daniel Golden, NASA Administrator, in his strategic outlook for 1999 (reference: http://hq.nasa.gov/office/nsp/outlook.htm) provides a statement of strategic intent for the agency. In this statement he outlines a three part mission in which “Technology Development and Transfer” is recognized as a cornerstone of the mission for the agency in the coming year. This is consistent with the 1999 external assessment (reference: http://hq.nasa.gov/office/nsp/assess.htm) in which the administration places priority on the promotion of “high technology for economic growth through effective partnerships”. The prime Space Flight Operations Contract, SFOC (reference: Contract NAS 9-20000) awarded to United Space Alliance, section G-14 and G-15 requires the contractor to provide a portion of the contract funds to small business and to support the “Government’s Technology Transfer Program”. These statements have served to emphasize the importance of exploring new ways for NASA and prime contractors to interface with industry to meet the goals implied. This new project development strategy is an example of how United Space Alliance is exploring new ways to increase the marketability of technology developed on the Space Shuttle Program. Particularly technology developed on upgrade projects for the benefit of small business in the commercial sector while decreasing Space Shuttle Program development and sustaining costs.

The status quo method employed to develop Space Shuttle ground systems software for upgrades or replacement projects is to develop custom code using commercial hardware and compilers. The new methodology explored here involves adopting a software application counterpart available in industry to provide baseline functionality and developing additional capabilities required for the upgrade or replacement system in collaboration with the software vendor. Utilizing this new development method leads one to the assertions that by leveraging technology available in industry initial software acquisition cost can be reduced, the time to market for the software product can be reduced and a superior product can be delivered to Space Shuttle operations. In addition, the technology developed should be closer to building on the state-of-the-art rather than reinventing the state-of-the-art, making the technology developed more valuable to industry and the American public.

In order to evaluate the merits of employing a collaborative approach to system development United Space Alliance selected a project that was well underway employing the status quo method to serve as a pathfinder. The Space Shuttle, Record and Playback

Subsystem (RPS) ground stations upgrade project was selected as the pathfinder. The project had been in development for approximately six years employing the status quo development methodology prior to it's selection.

This pathfinder project represents a stress test for evaluation of the method because the momentum behind software development activities underway will have to be overcome. The project defined a software development turnaround, for the purposes of evaluation, in terms of key project management metrics (i.e. cost, schedule and technical). This turnaround is defined as a 50% reduction in anticipated software development labor and schedule. The improvement in the technical metric was a little harder to quantify. Two surveys were performed in order to derive a comparative analysis between the two methods. Three key technical categories were defined and a turnaround was defined as a marked improvement in two of the three. The categories were marketability, software product quality and maintainability. A comprehensive set of attributes were developed for each category to aid in scoring. These sets of attributes are presented later in this paper. The first survey was designed to determine the relative importance of the attributes within each category for the software product. The second survey was performed after software development was complete and scored the delivered product against the most likely outcome of continued development on the custom code. The cost associated with the purchase of the COTS software was converted to equivalent manpower and factored in as additional software development resource available for custom code development. The results of these surveys were organized and presented in a Kepner Tregoe decision matrix for comparative analysis. A marked improvement was defined as a 100% improvement in absolute score within a category.

DEVELOPMENT METHOD AND VENDOR SELECTION PROCESS

Initially, a detailed estimate to complete the project assuming continued custom code development was performed which provided a baseline for evaluating the new methodology. This estimate to complete was based on a functional analysis of the custom code under development against the functional requirements for the upgrade. Functionality was divided into three categories for the purpose of this analysis; telemetric, station management and data product tools. Telemetric functionality was defined as the capabilities to acquire measurement data from the PCM or FM carriers, route this data to a location on the ground station, and also capture the data in a file for production of data products. Station management functionality was defined as the capabilities to setup and manage data acquisition as well as monitor real time elements status. Data product and tools were defined as the capability to process the raw data acquired into data products acceptable to the user. Figure 1 is a summary of this analyses. The entire pie in each category represents the functionality required for the upgrade in each category. The three slivers contained in the regions labeled "Functionality Addressed in the Custom Software"

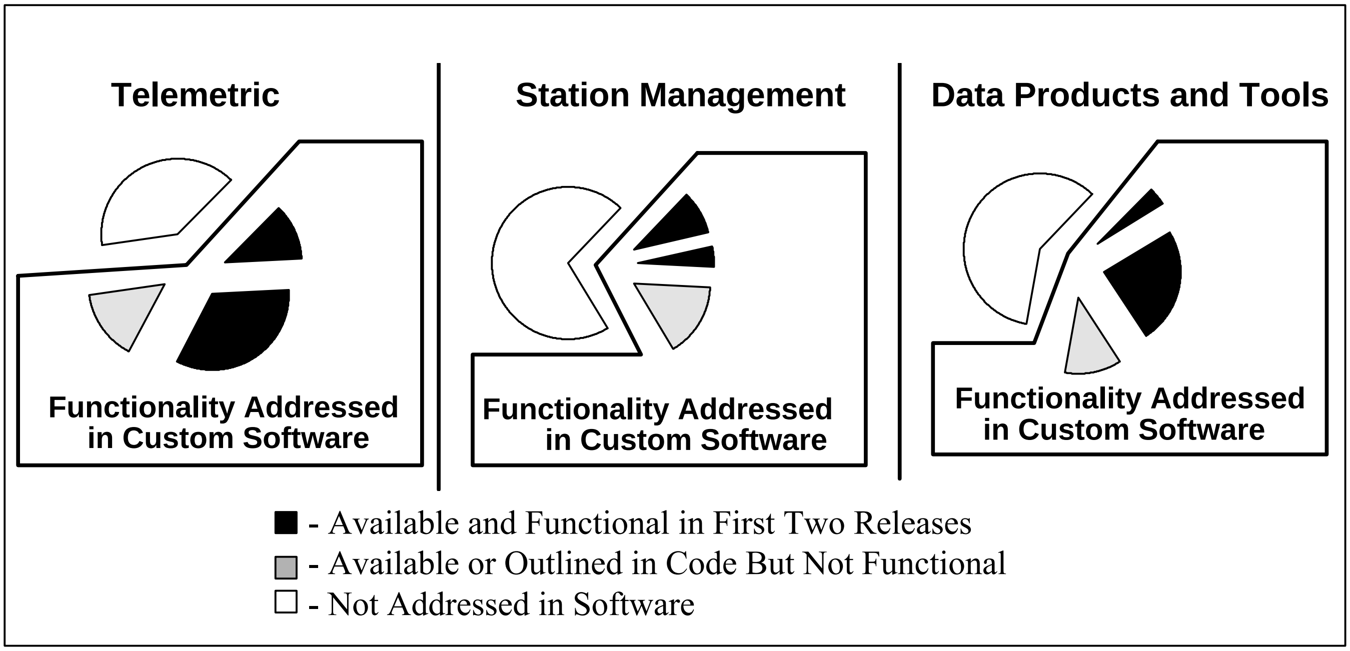

Figure 1 ***Evaluation of the Custom Code Revealed Large Gaps in Functionality***

represents capability outlined in the custom code. The two black slivers in this region represent the capabilities inherent in the first two releases of the custom code. The third shaded slice in this region represents capabilities that were outlined but not functional. The slices remaining outside the region represents capabilities requiring new development work to deliver. This means that approximately one third of the telemetric, three quarters of the station management and two thirds of the data products and tools capabilities required new development assuming continued development of the custom code. Project leadership and management derived an estimate to complete for the custom software. This estimate was based on a detailed knowledge of the functionality required, performance histories of the developers involved and developer's estimates. It was estimated that to complete the custom code would require an additional twenty man-years of productive software development effort expended over five years with a probable growth of twenty percent in both manpower and schedule.

An industry survey coupled with a series of product demonstrations were performed to determine if a Commercial Off The Shelf (COTS) product was available that met the functional requirements for the upgrade. The survey revealed that no commercially available software package met all the requirements of the upgrade. There were however several close functional matches including, Veda Systems, Huerikon, Harris, Metraplex, AP Data Systems, Avtec and Acromag. Hardware for the upgrade project had already been purchased and major change to the hardware architecture was seen as cost prohibitive. As product demonstrations and investigations progressed, it became apparent that general purpose telemetry software was fairly coupled to the hardware selected when

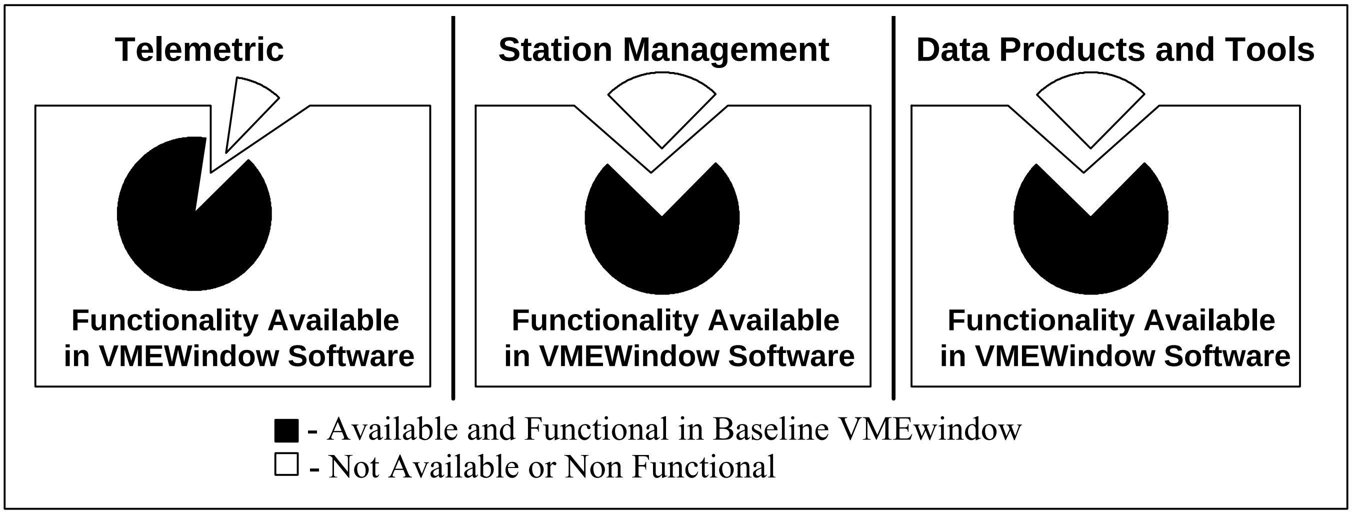

Figure 2 ***Evaluation of VMEwindow Revealed Much Smaller Gaps in Functionality***

development began. Of course, anyone can do a port but the challenges associated with developing additional capabilities concurrent with a port to a new hardware architecture was an approach deemed to have excessive technical risk. In addition, the vendor selected had to provide flexibility during development and licensing which will be discussed later. All of these factors lead to a single viable vendor and product combination. AP Data Systems wrote VMEwindow to a hardware architecture very similar to the one that was in storage for the upgrade and provided the flexibility to complete the project. A functionality assessment, similar to the one performed above, was performed in order to scope the software development effort required to complete the project. It was estimated that 10% of the telemetric, 25% of the Station Management and 25% of the Data Products and Tools capabilities would require new development. An aggregate 58% of the functionality was missing from the custom code and 20% from VMEwindow. An estimate of 7 manyears productive software development spent over 1.7 years to augment VMEwindow to meet the requirements of the upgrade can be derived given the estimate to complete the custom code. This is consistent with our definition of a software development turnaround and assumes that USA would develop the additional capabilities without collaboration. A presentation made to USA and NASA management in December 1998 resulted in an official go ahead approximately a month later.

COLLABORATIVE SOFTWARE DEVELOPMENT

Collaborative software development started the second week of March 1998. The first order of business was to train USA developers in the VMEwindow development environment. The systems engineering performed to identify the system level requirements aforementioned helped the development team identify gaps in functionality between the baseline VMEwindow product and the upgrade ground station. Although there were many minor modifications the collaboration resulted in 4 technology transfer, with innovators from USA and AP Data Systems. These were submitted to NASA for review. Software development was completed the second week in November 1998. These technological advancements represent modification to existing technology to improve and expand the capabilities of the baseline product. Figure 3 is a block diagram of the upgrade ground station architecture and will provide a reference in the following discussions on software developed and submitted for technology transfer. The hardware architecture consists of Sun workstations connected to multiple PCM and FM processing VME chassis which use reflective memory as a real time data transport mechanism. The chassis use a Motorola processors and SBS Berg Telemetry System cards. The software architecture consists of the VxWorks operating system, VMEwindow, Matlab, Dataviews, PVWave and new code developed as a result of the collaborative software development activities described above. The software products development for the project and submitted to NASA for technology transfer are described below.

SBUS SCRAMNET INTERFACE FOR VMEWINDOW

A reflective memory network shown in the center of Figure 3 is the primary data transport vehicle for raw data from the PCM Processors (PCMPs) to the Digital to Analog Programmable Converters, User and Station Controller workstations. The VMEwindow ground station telemetry package did not support an SBus SCRAMnet interface required to get data to the workstations. The challenge was to develop an application using the VMEwindow development environment that would provide the capability to acquire and display data at a single SBus workstation from multiple real time telemetry processors with deterministic and minimal latency. This code also had to be compatible and integrated within the VMEwindow environment providing the operator with a consistent interface. It was decided that a derivative work could be produced from the code already available for interface to the VME version of the SCRAMnet card. It was negotiated that AP Data System would perform code modification and initial development testing and USA provided the reflective memory topology, memory offset and data requirements in addition to concurrent code review, integration, testing.

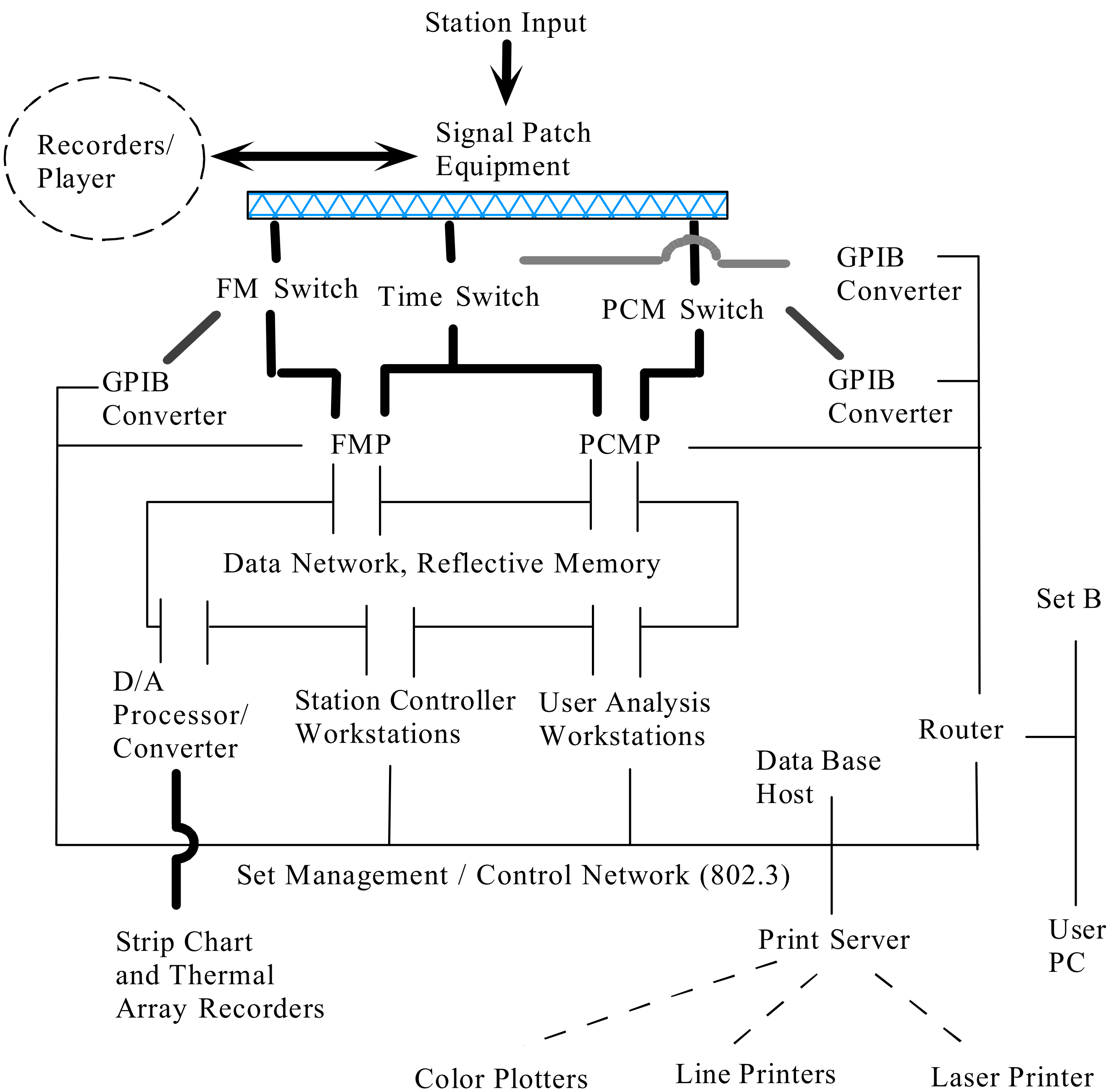

Figure 3 Ground Station Overview

STREAM DEFINITION FLAT FILE IMPORT CAPABILITY FOR VMEWINDOW

This software provides the capability to automatically load telemetry stream definition. This capability was seen as critical to the design because of the thousands of measurements that must be loaded to support each mission. Space Shuttle PCM down link, data location and unit information is contained in a data base. Baseline VMEwindow provides a manual utility to load this information into a stream definition but did not provide the capability to automatically load stream definition. The set of software products developed provide the capability to create an ASCII flat file from an SQL data base and import this flat file into the stream definition. This capability not only reduces time required to setup to support a mission but improves the reliability of the software load by cutting down on input errors. This product prompts for parameters to define a PCM stream and accepts lists of measurement names read from and existing file and accesses the data base to create an ASCII flat file. This file contains all of the information necessary to define PCM stream parameters and populate the stream definition. This software product then is used to import this flat file into the stream definition.

GPIB BOARD SETUP AND CONTROL FOR FM SNAPSHOTS AND CALIBRATION

This software provides the capability to setup and control the NI1014 General Purpose Interface Bus (GPIB) board for specialized Frequency Modulation (FM) functions. Although a generic GPIB interface capability was available in the baseline VMEwindow environment, it as not taylored to meet the requirements of the project. The system uses a Metraplex digital discriminator and a Keithly switcher. The setup and control software allows individual control of both the discriminator and the switch and automated control of report generation. The setup functions integrate into the ground station software as a newly developed VMEwindow icon. Board level control is provided using existing AP Data Systems driver software. These reports include calibrations and data snapshots. A FM test signal that shifts the frequency over the bandwidth to represent five levels between -5V to 5V is input to the digital discriminator. The calibration software provides an average of 5 samples at each of these data levels. Once the set is calibrated, a FM snapshot can be produced providing the average, minimum, and maximum values for a 100 samples of real time data.

DATEL 622 DIGITAL TO ANALOG DRIVER, SETUP, AND CONTROL

This software product provides the ability to produce thermal array charts of data with loss of signal event indicators. The driver and control software provides the capability to setup and control the Digital to Analog (D/A) conversion functions of the board. This capability was not available in the baseline VMEwindow environment. The setup functions integrate into the ground station software as a newly developed VMEwindow icon. The control software provides the capability to select data channels to be output, select event indicators for edge trace output, scale processing, and calibration functions. The calibration output is used to verify user setup. The driver software provides the low level hardware setup and control of the converter card.

COLLABORATIVE SUPPORT, DEVELOPMENT, AND LICENSING AGREEMENT

There are several challenges facing the successful implementation of a collaborative method on a project in support of Space Shuttle operations. These challenges stem from the perception that the ability to support operations is somehow compromised and substantially changed derivative or new software technology developed in the baseline environment is not transferable to industry. One of the arguments in favor of in house custom development of support software is that the developers are often the most qualified to troubleshoot and repair time critical bugs. A mechanism had to be developed to ensure that an equal or superior capability would be available after the upgrade. This issue is addressed by training a set of in house software developers to be certified VMEwindow developers and providing access to off hour vendor support during critical launch operations. Rights to the source code for the project was seen as pivotal for the application of this method. This was due to the volatility of small business in the field of software development and that in order to support a particular mission the baseline product may require modification with little or no potential commercial application. Both of these issues were resolved by acquiring a "Project Buyout License" for the upgrade. This eliminated the need to escrow source code with a third party and enabled ad hoc modification of the baseline product for use in the ground station without infringement concerns. Another challenge was that of combating obsolescence as an ongoing effort and minimizing in house configuration management requirements. This was addressed in the support agreement by acquiring special user and developer support. In addition to critical launch support mentioned above, Critical User Product Support is made available in which immediate technical attention is paid to a particular user software issue. A product upgrade cycle is defined in the agreement where the operators are provided changed user documentation and assistance installing the upgrade. The collaborations were so successful that special provisions were negotiated to ensure future endeavors would adopt a similar course. The systems engineering required to enable successful collaboration required that the interfaces and internal function points be defined to estimate the effort. Provisions are made to acquire systems engineering participation and assistance to this point. As mentioned earlier USA has interest in transferring technology to NASA under the Technology Transfer program particularly for the benefit of small business. Special provisions were negotiated to secure technology transfer rights because they not only meet the NASA goal of technology transfer but help lower configuration management costs as AP Data Systems adds these new capabilities into their baseline product.

FINDINGS

In order to derive a figure for comparison, the actual manpower expended to complete the software is added to a conservative equivalent manpower representing the additional cost incurred to procure the COTS software. The resulting figure is 6.6 man-years and when compared to the estimate to complete the custom code reveals a savings of 13.4 to 17.4 man-years. This figure represents savings in initial software development. When we conservatively convert these numbers to equivalent dollars we get more than $800K to $1M in software development labor savings.

The software was declared complete the second week in November 1998. This meant that the software development actually took eight months rather than the five to six years estimated to complete the custom code.

Table 1 presents a summary of a comparative analysis designed to determine the relative technical merit of the delivered software and the most probable outcome of having continued using the status quo development method. Expert engineering judgment is relied upon as the basis for this analysis in the absence of formal reliability and maintainability analyses. The first column of table 1 lists the categories defined to represent technical merit for the purposes of this analysis. Sub-indentured under each of the three categories (i.e. Marketability, Software Product Quality and Maintainability) are attributes of the categories. The second column represents the results of the first survey. This first survey was sent to manager level and above personnel involved in upgrade or replacement projects. The survey was designed to determine the relative importance of the attributes within a category from the perspective of the leadership. The responses from each of the respondents were normalized and an average value was calculated for each attribute. The third and fourth columns represent the results from the second survey. This survey was sent to members of the development team with knowledge of the functionality of the custom code, the track record of accomplishments during custom development and the functionality of the delivered code. The respondents were to provide a relative score on the delivered software and the most probable outcome of continued development on the custom code. The additional costs associated with the procurement of the COTS software and development was converted to equivalent manpower and the respondents were asked to make their estimate assuming these additional resources were brought to bear on the custom software development. A score of 5.5 was considered neutral. This means that a response of 7 on the delivered software represented a score of 4 for the most probable custom product. The fifth and sixth columns list the absolute scores for the custom and delivered product respectively. This figure is calculated by multiplying the average normalized weight for the attribute by the relative score for the attribute. These figures are totaled for each category to obtain an absolute relative score for each of the alternative methods. These relative scores are used for an order of magnitude relative comparisons and are rounded to the nearest whole number in this presentation.

Table 1 Method Comparison Matrix

1	2 Average Normalized Weight	3 Average Probable Custom Product Score	4 Average Software Product Delivered Score	5 Absolute Score Probable Costom Product	6 Absolute Score of Software Delivered
Marketability					
Value to Current Customer	7.78	1.67	9.33	13	73
Value to Other NASA Projects	4.70	2.50	8.50	12	40
Value to Industry	2.22	2.67	8.33	6	18
Value to Future Shuttle Customers	6.38	2.00	9.00	13	57
Value to American Public	3.92	4.17	6.83	16	27
Total =				60	215
Software Product Quality					
Usability	5.07	2.67	8.33	14	42
Reliability	8.26	1.83	9.17	15	76
Functionality and Versatility	4.57	1.83	9.17	8	42
Extensibility	2.09	1.50	9.50	3	20
Total =				40	180
Maintainability					
Training Programs	5.24	2.33	8.67	12	45
Availability of Trained Personnel	3.69	2.83	8.17	10	30
Ability to Enhance Software	5.55	1.67	9.33	9	52
Documentation	3.42	2.00	9.00	7	31
Configuration Management	7.09	3.58	7.42	25	53
Total =				64	211

CONCLUSIONS

Comparative observations are made between the actual project performance and the projected performance assuming continued development of the custom code. These analyses revealed that on the project used as the pathfinder for this study, cost and schedule where conservatively reduced by eighty six percent. This represents a savings of over $800K in software development labor. It also shortened the development schedule by more than four years. The technical evaluation of the product delivered summarized in table 1 revealed a marked improvement in every attribute within all three categories evaluated. Several additional benefits are discussed including technology transfer, continued collaborative enhancement of the product and the ability to combat obsolescence. The project exceeds the initial goal for a software development turnaround defined for the purpose of the pathfinder.

This methodology represents significant improvement over the status quo and should be evaluated for implementation on future and ongoing NASA software development projects. It is the authors considered opinion the results warrant forced application of this methodology where close counterparts are available in industry. Strong close technical counterparts for data warehousing, recording, retrieval, command, control, data monitoring, network traffic generation and system administration functions can be found in industry.

ACKNOWLEDGMENTS

The author gratefully acknowledges the following individuals and organizations: Jonathan Morsics, Steven Prenger, Steve Gills, Andy Mason, Richard Price, Tom Perez, Earl Johnson, Bruce Chamberlain, the Integrated Data Systems Management Team and the Record and Playback Subsystem Upgrade development team.

This work was performed under the auspices of the Space Flight Operations Contract (SFOC, NAS 9-2000).