NASA/CR-2005-213692

Knowledge Capture and Management for Space Flight Systems

John L. Goodman

United Space Alliance
Houston, Texas

October 2005

The NASA Scientific and Technical Information Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The STI Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at **http://www.sti.nasa.gov**
- E-mail your question via the Internet: help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Telephone the NASA STI Help Desk at (301) 621-0390
- Write to:

NASA STI Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2005-213692

Knowledge Capture and Management for Space Flight Systems

John L. Goodman

United Space Alliance
Houston, Texas

National Aeronautics and
Space Administration

Johnson Space Center
Houston, Texas 77058

October 2005

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

This report is also available in electronic form at http://ston.jsc.nasa.gov/collections/TRS

Introduction

The incorporation of knowledge capture and knowledge management strategies early in the development phase of an exploration program is necessary for safe and successful missions of human and robotic exploration vehicles over the life of a program (Fig. 1). Following the transition from the development to the flight phase, loss of underlying theory and rationale governing design and requirements occur through a number of mechanisms. This degrades the quality of engineering work resulting in increased life cycle costs and risk to mission success and safety of flight. Due to budget constraints, concerned personnel in legacy programs often have to improvise methods for knowledge capture and management using existing, but often sub-optimal, information technology and archival resources. Application of advanced information technology to perform knowledge capture and management would be most effective if program wide requirements are defined at the beginning of a program.

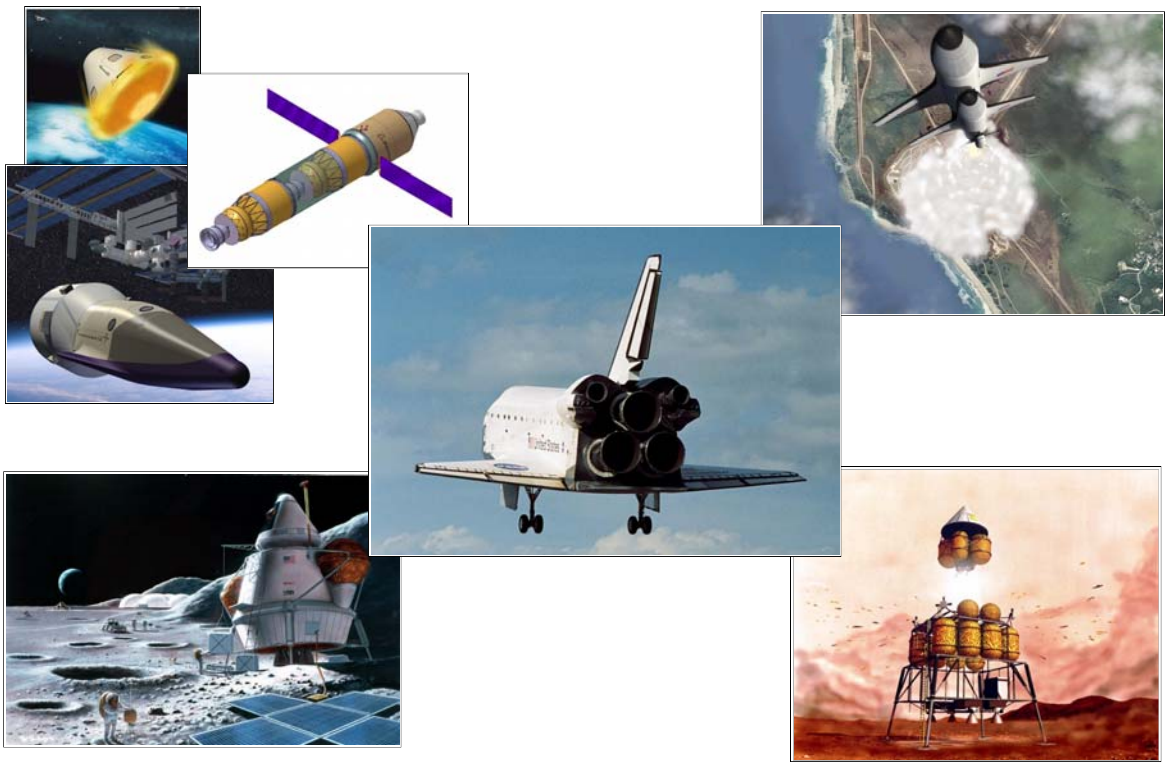


Figure 1. Knowledge capture will also be an issue for future programs. Lessons learned and best practices from the Space Shuttle can be applied to mitigate risk.

Knowledge Capture and Management Is Important

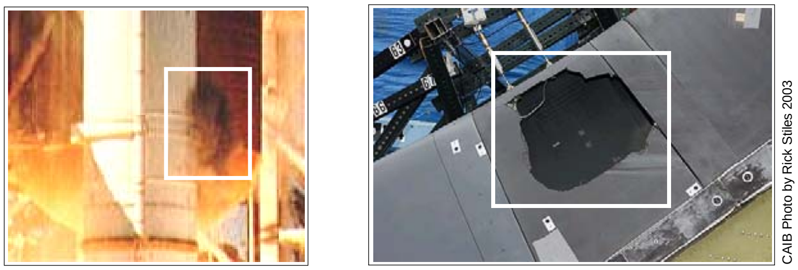
Meeting proposed autonomy and automation requirements in future exploration vehicles will require distributed computer systems and software of far greater complexity than those on the Space Shuttle or International Space Station (ISS). Ensuring safety and mission success depends on development, verification, performance analysis, and maintenance of hardware and software in on-board systems, ground systems, and ground facilities. Extensive analysis is performed in support of mission design, procedure development, and hardware evaluation. These activities require insight into underlying theory, requirements rationale, analysis techniques, systems performance and modification history, and software tools over the life of a program. The increasing complexity and proliferation of computer networks in on-board and ground systems necessitates insight into software design and operation, as evidenced by recent software related spacecraft accidents and the recovery of the Mars Exploration Rover *Spirit* from a software anomaly in early 2004.[1,2] An inadequate understanding of systems performance history was cited as a factor in the *Challenger* and *Columbia* accidents (Fig. 2).[3,4]

[1] Leveson, N. G., "The Role of Software in Spacecraft Accidents," AIAA Journal of Spacecraft and Rockets, Vol. 41, No. 4, July-August 2004, pp. 564-575.

[2] Reeves, G., and Neilson, T., "The Mars Rover Spirit FLASH Anomaly," 2005 IEEE Aerospace Conference, IEEE, New York, NY, 2005.

[3] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, U.S. Government Printing Office, Washington, DC, June 6, 1986.

[4] *Columbia Accident Investigation Board Report*, Volume I, U.S. Government Printing Office, Washington, DC, August 2003.



a) Smoke From Solid Rocket Booster Joint During *Challenger* Launch

b) Reinforced Carbon Carbon Leading Edge Damage After Foam Impact Test During *Columbia* Investigation

Figure 2. Inadequate Understanding of Systems Performance History Was a Factor in the Loss of Two Space Shuttles and Their Crews

The need for program wide, precisely defined knowledge capture and management strategies can best be discerned by examining challenges faced by engineering and management personnel working in legacy flight programs. Detailed knowledge of underlying theory and rationale governing design and requirements exists during the development phase of a vehicle and supporting ground systems. In the years following the transition from development to flight, corporate knowledge loss occurs through a variety of mechanisms. Over the last 45 years, U.S. human spacecraft have become increasingly complex (Fig. 3). Larger numbers of engineers are

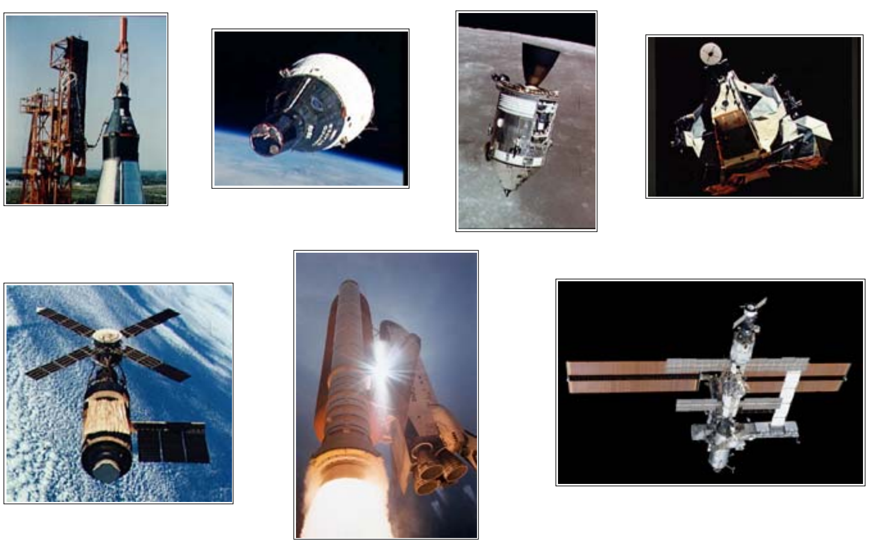


Figure 3. Knowledge Capture and Management Becomes More Challenging Due to Increasingly Complex Missions and Complex Spacecraft

required to understand, evaluate, and maintain increasingly complex spacecraft. Some engineers now joining the Shuttle Program were born after the first flight of the Space Shuttle in April of 1981. Longer program lifetimes increase the risk of corporate knowledge loss (Fig. 4, Table 1). The need for knowledge capture and management in the face of changing workforce demographics has been raised by the Columbia Accident Investigation

Board, the Aerospace Safety Advisory Panel, and the Government Accountability Office.[4-6] United Space Alliance is identifying cultural changes needed; is conducting surveys of existing knowledge capture and management mechanisms, and is proceeding with additional knowledge capture and management policy definition to ensure safety of flight through the end of the Shuttle Program.

[5] *Aerospace Safety Advisory Panel, First Quarterly Report*, February 2004.

[6] *Space Shuttle – Actions Needed to Better Position NASA to Sustain Its Workforce Through Retirement*, GAO-05-230, United States Government Accountability Office, March 2005.

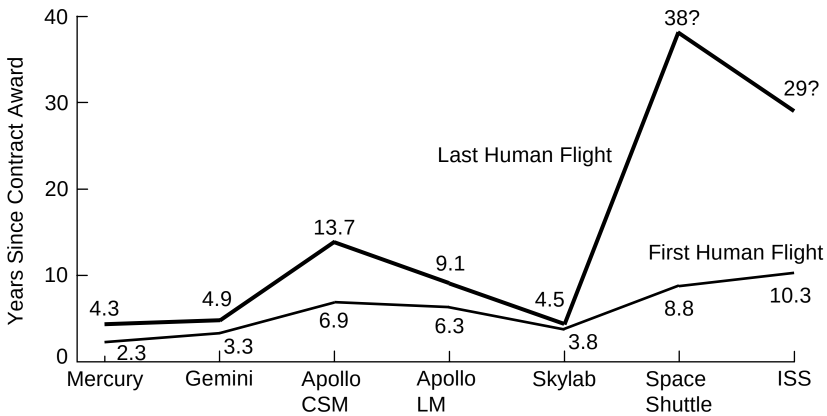


Figure 4. Elapsed Time From Prime Contract Award to First and Last Human Flights is Increasing

Table 1. Chronology of NASA Human Space Vehicles

Spacecraft	Prime Contract	First Human Flight	Last Human Flight	Time From Contract Award to First and Last Human Flights
Mercury	McDonnell Jan. 9, 1959	Freedom 7 May 5, 1961	Faith 7 May 15, 1963	2.3 years / 4.3 years
Gemini	McDonnell Dec. 22, 1961	Gemini 3 April 23, 1965	Gemini 12 Nov. 11-15, 1966	3.3 years / 4.9 years
Apollo Command/ Service Module	North American Nov. 28, 1961	Apollo 7 Oct. 11-22, 1968	Apollo-Soyuz July 15-24, 1975	6.9 years / 13.7 years
Apollo Lunar Module	Grumman Nov. 7, 1962	Apollo 9 March 3-13, 1969	Apollo 17 Dec. 7-19, 1972	6.3 years / 9.1 years
Skylab	McDonnell-Douglas Aug. 8, 1969[a]	Skylab 2 May 25, 1973 – June 22, 1973	Skylab 4 Nov. 16, 1973 – Feb. 8, 1974	3.8 years / 4.5 years
Space Shuttle	North American Rockwell July 26, 1972	STS-1 April 12-14, 1981	2010?[d]	8.8 years / 38 years[d]
International Space Station[b]	Sept. 28, 1988[c]	Flight 2A (STS-88) Dec. 4-15, 1998	2017?[d]	10.3 years / 29 years[d]

[a] Contract for primary and back-up dry workshops.
[b] Freedom (U.S., Canada, ESA, Japan) redesigned to International Space Station (including Russian elements) in 1993.
[c] Award of Work Packages 1, 2, 3 and 4 (Boeing, McDonnell-Douglas, GE Astro Space and Rocketdyne). Packages 1, 2 and 4 novated under Boeing on August 17, 1993. Work Package 3 was canceled in February 1991.
[d] Assuming Shuttle retirement in 2010 and ISS retirement in 2017 (A Budgetary Analysis of NASA's New Vision for Space Exploration, Congressional Budget Office, September 2004).

Corporate knowledge loss negatively impacts the ability of engineers to perform accurate analyses in a timely manner. Significant amounts of time may be expended in an attempt to understand analyses performed and technical decisions made in the past. In some cases, lack of insight may force an analysis to be completely redone. Incomplete understanding of system requirements rationale, underlying design theory, and systems performance history degrades the quality of engineering work. Corporate knowledge loss also makes it difficult for engineers to understand, evaluate, modify and reuse software years or decades after it was written and certified. The same is true for hardware and ground facilities. The result is increased life cycle costs and risk to safety and mission success. Effective mentoring and access to key historical documentation for second, third, fourth, and subsequent generations of engineers is critical in an industry with a turnover rate and little margin for error.

Organizations that embrace knowledge capture and knowledge management can be more effective at avoiding technical, cost and schedule risks over the life of a program. They also embrace a culture that values theoretical understanding, intellectual curiosity, and performance investigation, which enables information technology to be leveraged in an efficient manner to address the knowledge capture and management problem.

Why Knowledge May Not Be Captured or Accessible

Corporate knowledge loss occurs over time, and the effect can be difficult to detect. Over a time span of little more than a year, enough knowledge can be lost to complicate the investigation of recent performance analysis, anomaly resolution, and development efforts. Early in a program, people may be motivated to document new applications of technology in terms of requirements rationale, analysis, issue resolution, best practices, and lessons learned. Experience has shown that later in a program personnel may not be as motivated to create archival documentation for certified systems and processes.

Knowledge capture and knowledge management may be valued in an organization, but it may not be a demonstrated priority. Some projects do not routinely document requirements rationale, empirically or theoretically derived results, and performance history. Nor do they ensure that existing documentation containing this information is preserved in a manner that allows location and retrieval in the future. The decision is left up to each organization. Some will choose to document what they have done, others will not.

Not all engineers and managers are willing to take the time to document what they do in a manner that preserves knowledge. Creating formal or informal documentation takes time and good writing skills. The importance of knowledge capture for ensuring the technical competence of future personnel, or facilitating future software maintenance and reuse, is often overlooked. However, knowledge capture and management tends to be valued by engineers and managers who are responsible for the performance of a legacy system. Constraints such as schedule and available resources can also prevent project results from being properly documented.

Equations for inclusion in software requirements documents and software change requests are normally the formal software project deliverables. Formal or informal documentation of the development of theoretical or empirical equations and supporting analysis are not usually considered to be deliverables. While this information does exist at some point in the software life cycle, such documentation containing key knowledge and history may not be resident in an archival system that permits later location and retrieval by personnel that were not associated with the original development effort. Even if an engineer or manager chooses to formally document historical design rationale and theory, the documentation may not be easily accessible to other engineers in the program at the time of development or in the future. The requirement for marking documentation as proprietary (by companies) or export controlled (by federal law) makes retrieval of critical design information even more difficult.

Human space flight programs encompass many government and contractor organizations. Personnel who designed and developed the spacecraft are not typically in the same organization as those who perform mission planning, ground facility operation and maintenance, and real-time flight functions. This complicates the knowledge capture and management problem. In contrast, some robotic exploration missions are of short duration and small enough in terms of size of the work force that many of the engineers, scientists and software specialists that participated in requirements definition, hardware development, and software development also participate in the flight phase. This results in direct application of in-depth systems knowledge to mission planning, mission execution, and issue resolution.

Legacy Vehicles and Systems

Key knowledge for legacy vehicle on-board and ground systems exists at some point in time, either electronically, on paper, or in the heads of management and technical personnel. Much critical information may already be captured and accessible through existing processes and archival repositories. However, some knowledge may never have been preserved in a manner that makes it accessible to future engineers. In a repeated flight environment, where a product mentality may exist, it is necessary that technical personnel strive to maintain a culture of intellectual curiosity and pursue a deeper understanding of requirements rationale, systems design, and systems performance history.

A. The Education of an Engineer

In the mid-twentieth century, it was normal for an engineer or manager to have worked on at least half a dozen different projects (aircraft, missiles, spacecraft and their associated ground support systems and facilities) before they were forty years of age. A variety of development experiences provided excellent opportunities for managers and engineers to develop technical and problem solving skills. By the 1970s, the number of aerospace projects had decreased and projects lasted decades, decreasing the opportunities for personnel to develop their skills.

Most of the personnel who develop systems requirements, formulate the theory behind software algorithms, and perform critical analysis for flight techniques development and systems performance evaluation eventually leave a program. Over time, the responsibility of ensuring that vehicle and ground systems will perform as required rests largely with personnel who did not participate in the development phase of the program. While many personnel in a program are conversant in requirements and maintenance processes, they do not always possess the design insight of their predecessors. Engineers and managers who have spent their careers in a flight environment may not have had the same opportunities to develop theoretical and analytical skills as those who worked in the program development phase. Historical documentation plays an important role in educating later generations of personnel in theoretical and data analysis techniques. Design documentation is needed not just to provide details of what a system does and why it was designed that way, but also to educate the future engineer or software specialist in the field.

B. Publicly Available Sources of Knowledge

Much information useful in the education of an engineer or software specialist can be found in the open literature in the form of program histories, textbooks, short courses, journal articles and conference papers. However, as open literature sources tend to be general in nature, they will not contain more specific, equation level details and design rationale, particularly for applications involving proprietary software and security concerns. Limitations in length and content relegate publicly available sources of information to a supplementary role. Comprehensive internal documents are the primary vehicle for preserving knowledge.

1. Program Histories

Histories of programs and case studies can provide important background information on high-level systems requirements and how they were shaped by political, policy, business, and budgetary considerations.[7-11] Significant engineering challenges and operational histories may also be covered in open literature publications.[8-11] This is important background information for legacy system personnel and can aid in identifying more specific areas for technical study and in forming lines of investigation when interviewing veteran engineers and managers.

2. Textbooks, University Classes and Professional Short Courses

Textbooks by academics or specialists in industry are a valuable source of fundamental principles behind mathematical theory, analysis, and systems design. However, publishing considerations (cost, book size, potential market) place limitations on the amount of material that can be covered. Furthermore, works by academics may not include operational considerations that heavily influence the design and operation of vehicle and ground systems. Texts written by participants in the real-world application of technology can provide excellent examples that enhance reader comprehension of the theory. Equations in a text must be verified before they can be introduced into detailed software requirements. In the interest of keeping page count to a minimum and to lower the risk of introducing errors into a text, equation derivations do not include intermediate steps. This can significantly complicate attempts to verify the equations. While “an exercise left to the student” is appropriate in the university environment, time consuming verification of theoretical results can lead to cost and schedule concerns on a project in industry.

Continuing education is important for maintaining and enhancing the skills of the engineering workforce. Short courses are particularly useful for gaining an understanding of new technology about to be introduced into a legacy system. Short courses concerning Global Positioning System and strapdown navigation were invaluable to Johnson Space Center (JSC) personnel applying these technologies to the Space Shuttle, ISS, and X-38 vehicles.

3. Journal Articles and Conference Papers

Articles and conference papers written by those who participated in the design, development, and flight phases of a vehicle can provide insight into

[7] Aronstein, D. C., Hirschberg, M. J., and Piccirillo, A. C., *Advanced Tactical Fighter to F-22 Raptor: Origins of the 21st Century Air Dominance Fighter*, AIAA, Reston, VA, 1998.

[8] Heppenheimer, T. A., *Space Shuttle Decision, 1965-1972* (History of the Space Shuttle, Volume 1), Smithsonian Institution Press, Washington, DC, 2002.

[9] Heppenheimer, T. A., *Development of the Shuttle, 1972-1981* (History of the Space Shuttle, Volume II), Smithsonian Institution Press, Washington, DC, 2002.

[10] Piccirillo, A. C., and Aronstein, D. C., *The Lightweight Fighter Program: A Successful Approach to Fighter Technology Transition*, AIAA, Reston, VA, 1997.

[11] Jenkins, D. R., *Space Shuttle – The History of the National Space Transportation System – The First 100 Missions*, Specialty Press Publishers, North Branch, MN, 2001.

programmatic requirements, the evolution of design concepts, background information on the application of a theory or technology, and rationale behind system operation during flight and lessons learned.[12-15] Papers can detail how a theoretical result was used in a system and how the algorithm performed in flight.[16,17] The high level discussion of programmatic design considerations and vehicle design aspects is useful to newer engineers striving to understand a legacy system.[18-22] Another excellent example is a book covering the design and rationale behind the Shuttle avionics architecture, written by two of the original designers.[23]

C. Internal Sources of Knowledge

Formal and informal documentation within a program varies widely in terms of quality, scope, and accessibility. Knowledge in the form of presentations and technical reports, informally maintained reference books, procedures, requirements documents, and derivations of equations may be adequately captured and archived through existing processes. However, there also may be serious deficiencies in documentation quality, scope and preservation.

1. Presentations and Technical Reports

Well-written presentations and technical reports are a valuable source of information. Test and anomaly resolution reports, although they may not contain derivations and theoretically or empirically derived constants, can contain useful information on the architecture, design rationale and algorithms. Lengthy briefings may contain large amounts of data and bullet points, but lack prose, historical context, derivations of equations and explanations leading to an understanding of theory, requirements rationale, and operational constraints. In addition, many reports and presentations are not written so that years later someone will be able to understand them and place the material in context. Many organizations within a program create training and job certification materials. These play a valuable role, but may not provide answers to many engineers' questions about requirements rationale, systems performance, systems history, and the theory behind software algorithms.

2. Software Requirements Documentation

Configuration controlled software requirements documents are invaluable in helping personnel understand software functionality. While such documents contain equations and logic implemented, they rarely provide insight into how the equations were derived, how values of constants were determined, or references to other sources (books, external or internal papers) used for theoretical development. This information, while valuable to the practicing engineer and software specialist, is often outside the scope of a software requirements document. Questions about requirements rationale frequently come up in meetings where software functionality, performance, and proposed changes to software are discussed. Existing documents often do not contain the answers to these questions.

3. Derivations of Equations

Preserving internal documentation concerning derivations is important, as many mathematical results used in software often do not exist in the open literature. A common question among those working with legacy software is "Where did those equations come from?" An understanding of the theoretical or empirical development of equations and constants appearing in safety-critical or mission-critical software is essential if an engineer is to properly evaluate software performance, perform modification, or re-use the

[12] Young, K. A., "Representative Space Shuttle Missions and Their Impact on Shuttle Design," AIAA/ASME/SAE Joint Space Mission Planning and Execution Meeting, AIAA, Reston, VA, 1973.

[13] Kachmar, P. M., and Wood, L., "Space Navigation Applications," Journal of the Institute of Navigation, Vol. 42, No. 1, Institute of Navigation, Fairfax, VA, 1995.

[14] Goodman, J. L., "Space Shuttle Navigation in the GPS Era," Proceedings of the Institute Of Navigation National Technical Meeting, Institute of Navigation, Fairfax, VA, 2001.

[15] Goodman, J. L., "Application of GPS Navigation to Space Flight," 2005 IEEE Aerospace Conference, IEEE, New York, NY, 2005.

[16] Harpold, J. C., and Graves, C. A., "Shuttle Entry Guidance," Journal of the Astronautical Sciences, Vol. 17, No. 3, July-September 1979.

[17] Harpold, J. C., and Gavert, D. E., "Space Shuttle Entry Guidance Performance Results," AIAA Journal of Guidance, Control, and Dynamics, Vol. 6, No. 6, AIAA, Reston, VA, November-December 1983, pp. 442-447.

[18] Johnson, C. C., and Petynia, W. W., "Design and Programmatic Philosophies Revealed in Apollo Spacecraft Design and Development," AIAA Annual Meeting and Technical Display, AIAA, Reston, VA, 1981.

[19] Boynton, J. H., and Kleinknecht, K. S., "Systems Design Experience From Three Manned Space Programs," Journal Of Spacecraft And Rockets, Vol. 7 No. 7, 1970, pp. 770-784.

[20] Love, E. S., "Advanced Technology and the Space Shuttle," AIAA 9th Annual Meeting and Technical Display, AIAA, Reston, VA, 1973.

[21] Scott, H. A, "Space Shuttle Orbiter Configuration Case History," AIAA Aircraft Systems and Technology Conference, AIAA, Reston, VA, 1978.

[22] Thompson, R. F., "The Space Shuttle – Some Key Program Decisions," AIAA 22nd Aerospace Sciences Meeting, AIAA, Reston, VA, 1984.

[23] Hanaway, J. F., and Moorehead, R. W., *Space Shuttle Avionics System*, NASA SP-504, NASA, Washington, D.C., 1989.

software. Reverse engineering the theoretical development of equations from software requirements documents can be an arduous and time-consuming task. The fundamental theory behind equations may be based on open literature sources, but the actual implementation may appear to be different from that in the literature and be accompanied by code that reflects implementation specific considerations.

One such example concerns Lambert's theorem, which has been used in many spacecraft applications in onboard or ground software to compute the velocity required for orbit adjustment maneuvers.[24,25] Numerous solutions have appeared over the last 200 years, a few of which are depicted in Fig. 5. All solution methods involve the solution of transcendental equations through iterative means. Actual implementation of a Lambert algorithm in ground or on-board software may be based on work appearing in the open literature. However, many aspects of the Lambert algorithms actually in use may appear to be different from published theoretical developments. These differences are due to considerations for specific mission applications, software error handling, choices of numerical iteration algorithms, choices of convergence criteria, and software interfaces. An understanding of application specific design considerations and constraints is essential for an engineer to understand the theoretical development, requirements implementation, and algorithm performance. Without this understanding, it may be impossible for an engineer to recreate the theoretical development of the algorithms starting with published derivations and ending with the final form as implemented in software. This understanding is critical during software verification, anomaly investigation, anomaly resolution, and if the algorithm is to be ported to a new application with different requirements than the original.

$$\sqrt{\mu}\,(t_2-t_1)=a^{\frac{3}{2}}\left[(\alpha-\sin(\alpha))-(\beta-\sin(\beta))\right]$$

a) Lagrange (1778) [25]

$$y^3-y^2=m^2\left[\frac{2\psi-\sin 2\psi}{\sin^3\psi}\right]$$

b) Gauss (1809) [25]

$$\sqrt{\mu}\,t=\left[\frac{s}{C(x)}\right]^{\frac{3}{2}}S(x)\mp\left[\frac{s-c}{C(y)}\right]^{\frac{3}{2}}S(y)$$

c) Battin (1964) [26]

$$N=n_s\,t=\frac{1}{z|z|^{1/2}\,2^{1/2}}\left[\frac{1-k}{2}m\pi+ k[\,f(|z|^{1/2})-|z|^{1/2}(1-z)^{1/2}]- [\,f(w|z|^{1/2})-w|z|^{1/2}(1-w^2 z)^{1/2}]\right]$$

d) Gedeon (1965) [27]

$$T=\phi(u)(-E)-q\,K\,\phi(u)(-KE)$$
$$\phi(u)=\frac{2}{u^{3/2}}\left[\sin^{-1}u^{1/2}-u^{1/2}(1-u)^{1/2}\right]$$

e) Lancaster, Blanchard and Devaney (1966) [28]

$$2\,\beta=\frac{1}{\cos\theta_1\left(C_1\cos\theta_1+C_2\sin\theta_1\right)}=\frac{V_1^2}{\mu}R_1$$

f) Okhotsimsky (1968) [29]

$$2\,\omega\,t=b_1\left(1-\cos E\right)+b_2\left(E+n\,2\,\pi\right)+b_3\sin E$$

g) Jezewski (1975) [30]

$$t_f=\frac{|r_1|}{V\cos\gamma}\left[\frac{\tan\gamma\,(1-\cos\theta_f)+(1-\lambda)\sin\theta_f}{(2-\lambda)|\,r_1\,|\,/\,|\,r_2\,|}+\frac{2\cos\gamma}{\lambda\left[(2/\lambda)-1\right]^{3/2}}\tan^{-1}\frac{\left[(2/\lambda)-1\right]^{1/2}}{\cos\gamma\cot(\theta_f\,/\,2)-\sin\gamma}\right]$$

h) Nelson and Zarchan (1992) [31]

Figure 5. A Few of the Many Versions of Lambert's Equation

[24] Volk, O., "Johann Heinrich Lambert and the Determination of Orbits for Planets and Comets," Celestial Mechanics and Dynamical Astronomy, Vol. 21, pp. 237-250.

[25] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Edition, AIAA, Reston, VA, 1999.

[26] Battin, R. H., *Astronautical Guidance*, McGraw-Hill Book Company, New York, NY, 1964.

[27] Gedeon, G. S., "A Practical Note on the Use of Lambert's Equation," AIAA Journal, Vol. 3, No. 1, January 1965, pp. 149-150.

[28] Lancaster, E. R., Blanchard, R. C., and Devaney, R. A., "A Note on Lambert's Theorem," Journal of Spacecraft and Rockets, Vol. 3, No. 9, AIAA, Reston, VA, September 1966, pp. 1436-1438.

[29] Okhotsimsky, D. E., *Dynamics of Space Flight*, Moscow State University, Moscow, Russia, 1968.

[30] Jezewski, D. J., "K/S Lambert Problem," Society of Engineering Science Annual Meeting, Society of Engineering Science, 1975.

[31] Nelson, S. L., and Zarchan, P., "Alternative Approach to the Solution of Lambert's Problem," Journal of Guidance, Control and Dynamics, Vol. 15, No. 4, AIAA, Reston, VA, July–August 1992, pp. 1003-1009.

4. Existing Databases and Archives

Numerous databases and archives (electronic or hardcopy) may exist in a program. These are a valuable resource for published documents that contain knowledge and systems history. However, the location and relevant contents of formally maintained archives is often difficult to ascertain due to the myriad of databases, organizations, and contracts that exist across a flight program or a government agency. Technical and management personnel may encounter difficulty in searching for relevant documentation, assuming that they are aware of the existence of the archive in the first place. Furthermore, while the archive itself may be well maintained, the placement of quality documentation in the archive may not be consistently practiced across the program and over time.

Unfortunately, the underlying material associated with these archived documents is mostly contained in informal, personal repositories, such as engineers' paper files, computer hard drives, or other electronic media. These informal repositories are generally not documented, have little or no procedural controls, and are often discarded when personnel retire, re-organizations occur, or when it is perceived that the material is no longer needed.

D. Examples of Knowledge Capture in a Legacy Program

Knowledge capture and knowledge management efforts in a flight program may be performed many years after the development phase was completed. However, much knowledge may have already been lost. Conscientious personnel desire to document what they are doing while the task is underway, and ensure that others know the documentation exists. Personnel conducting such efforts may be constrained by cost, schedule, and process limitations that constrain approaches to low tech, low impact methods. Individual efforts by concerned engineers and managers can capture valuable knowledge that is not already captured formally, or can create new documentation covering subjects that are not adequately documented. However, these personnel may have little ability to improve the state of knowledge capture and management across a program or advocate changes to existing processes.

While collecting and preserving historical documents on a small scale can be accomplished by an individual manager or engineer with little or no impact to cost and schedule, developing new organizational or program wide methods and systems for knowledge capture and management is more difficult. Organizational inertia can make it difficult to change processes so that critical knowledge is identified, captured, and archived. Creating new documentation can impact cost and schedule. Procuring and integrating new information technology software can be expensive and difficult due to organizational inaction, information technology policies, and network architecture considerations. Existing libraries, archives, and software applications may have to be used for knowledge management.

Research into knowledge management theory is important for the development of new information technology applications and the integration of such technology into the workplace. However, such research is often difficult for the practicing engineer or manager in a legacy program to apply, as it is of a theoretical and abstract nature (Fig. 6).

This section contains examples of knowledge capture and management efforts from the Shuttle Program. They are included to promote creative thinking to identify and define low cost, low overhead knowledge capture and management efforts. Of the examples given, only the Engineering

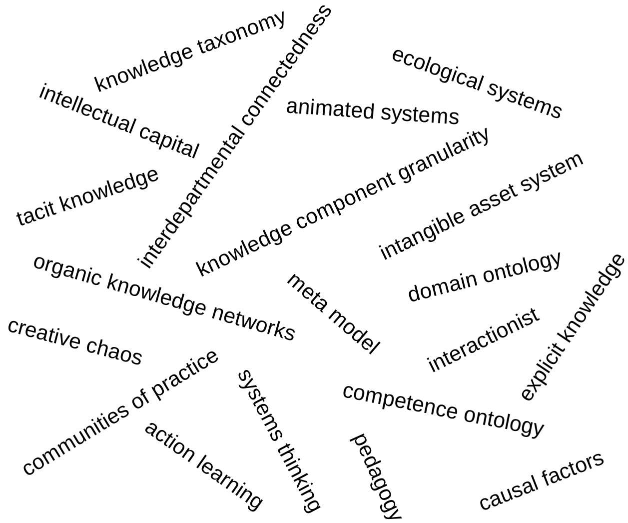


Figure 6. While Important, Knowledge Management Theory Can Be Difficult for Personnel to Leverage

Knowledge Base required a small number of dedicated personnel and the purchase of new hardware. The web-based knowledge capture and management tool for Mission Control software development is the only example that required the purchase of new software.

1. Collecting and Making Historical Documentation Accessible

Younger engineers are often placed into the role of historians and detectives in order to locate, identify, and acquire key historical documents. This is not always an easy task, especially since an engineer may not be aware of what documents and memos were written. Many newer engineers may not know who the experts were during the development phase, nor do they know where to look for extant material. Organizational or individual self-interest and group barriers can also prevent engineers from obtaining key historical documentation, even if it is non-proprietary and government owned. Many veteran engineers possess or are aware of, memos, white papers, test reports, and presentations that provide valuable insight into requirements rationale, theory, and systems performance history. These materials may not have survived in the originating organization, but are still kept by others, or may have had limited distribution in the first place.

On January 17, 1996, a crack (accompanied by a loud bang) developed in the Mobile Launch Platform (MLP) deck while the crawler was transporting the MLP and the shuttle *Discovery* from the Vehicle Assembly Building to launch Pad 39A for mission STS-82 (Fig. 7). The transport operation was stopped until the issue could be evaluated. An engineer had maintained a history of structural deflection measurements in a brain book[32], which was stored in a truck. Data from the brain book was used to quickly determine that the crack did not pose a risk to flight hardware or personnel and the rollout to Pad 39A was resumed.[33]

This incident, along with Shuttle vehicle and associated ground hardware experience during the Shuttle Program, led United Space Alliance Ground Operations engineers at the Kennedy Space Center (KSC) to develop the computer-based Engineering Knowledge Base (EKB). The EKB (Fig. 8) consists primarily of Commercial Off-The-Shelf Software (COTS). It captures and archives data that may not already be formally retained (engineering notes, studies, analyses, calculations, lessons learned, engineering brain books, etc.) through previously existing mechanisms. Paper documentation is scanned for inclusion in the knowledge base. The EKB permits timely

[32] A brain book is a personal encyclopedia that an engineer compiles for reference purposes.

[33] "Mobile Launcher Platform Develops Crack During STS-82 Rollout," NASA Press Release 97-13, NASA Kennedy Space Center, January 17, 1996.

a) Crawler transporting *Discovery* and the MLP to the Launch Pad

b) Crack in the MLP

c) Close-up of Crack

Figure 7. Structural Deflection History From a Notebook Stored In a Truck Was Instrumental In Resolving a Crack Issue That Occurred During the STS-82 Rollout

access to historical information and facilitates the capture of such documentation as it is created. Overt solicitation of engineering material from technical personnel has been effectively conducted. The EKB has made it easier for engineers to do their jobs, which has motivated personnel to contribute material to it. Former employees who return as consultants are able to retrieve their notes and other material that is important for their consulting tasks. A key to the success of the EKB was in-depth study of the KSC work culture by veteran KSC personnel.

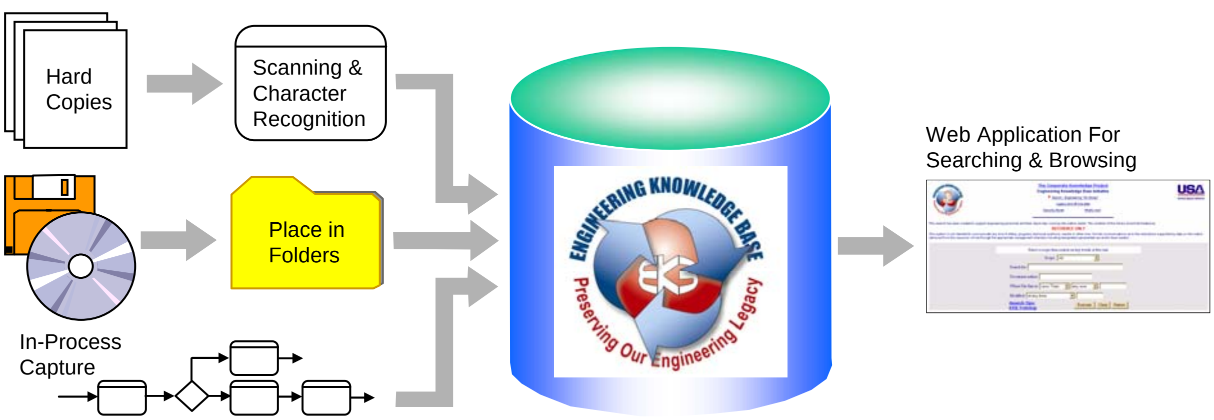
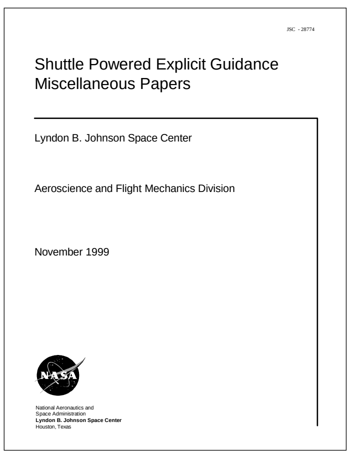


Figure 8. As of July 26, 2005, the Engineering Knowledge Base at the Kennedy Space Center Contained 144,636 Items, Consuming 127 Gigabytes of Memory

To create the EKB, existing software applications were used, and a small amount of Visual Basic code was written. A dedicated server was procured to meet growing storage needs. A harvester collects and screens material from donors. A small team of contract personnel performs scanning, loads the electronic documents into the EKB, and burns a compact disk with electronic versions of the hardcopy material. The harvester returns the original material (hardcopy or electronic) to the donor along with the compact disk.

Concerned JSC engineers in the Shuttle Program have contacted many veteran personnel, who participated in algorithm and requirements development, for information on rationale and derivations or for copies of old memos and presentations. Cooperative veterans have done their best to answer questions and have provided access to old files. However, veteran personnel may forget what documentation they have in their possession and may not recognize material as being significant to younger, intellectually curious engineers.

Several engineers at NASA JSC have taken steps to preserve key historical memos in the existing JSC technical library system. About 1,500 pages of historical, analysis and theoretical material on the Shuttle on-board rendezvous software application was compiled into three documents.[34-36] The material covered a thirty-year period, starting in the early 1970s. Much of the material, collected over a 15-year period, was not previously available to engineers who are concerned with the software application. This material proved to be of value to recently hired engineers even before the final versions of the compilations were published. The volumes were distributed to organizations within the Shuttle Program and were placed in the NASA JSC technical library. A similar effort was conducted by NASA personnel designing guidance software for the X-38 Crew Return Vehicle, which used theory originally developed for the Space Shuttle.[37] X-38 engineers published historical documents they had located in a compilation and placed that volume in the JSC technical library (Fig. 9a).[38]


JSC - 28774

Shuttle Powered Explicit Guidance
Miscellaneous Papers

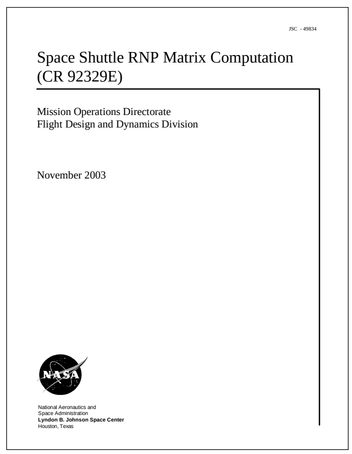
Lyndon B. Johnson Space Center

Aeroscience and Flight Mechanics Division

November 1999

National Aeronautics and
Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

a) Compilation of Memos on Shuttle Guidance


JSC - 49834

Space Shuttle RNP Matrix Computation
(CR 92329E)

Mission Operations Directorate
Flight Design and Dynamics Division

November 2003

National Aeronautics and
Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

b) Background Information on a Shuttle Software Improvement

Figure 9. Concerned Engineers Have Created New Documentation, and Placed Them in an Existing Technical Library For Preservation

2. Creating Specific Knowledge Capture Documents

Primarily in the 1970s, during initial development of Shuttle requirements and software, many internal technical reports were written detailing theoretical development of algorithms destined for use on-board the Shuttle and in supporting ground systems.[39-46]

Concerned engineers within the Shuttle Program, with management support, have created documentation to provide detailed supporting information for three recent Shuttle software changes.[47-49] The documents detail the derivation of equations and the theoretical and empirical basis for the requirements change, as well as the evolution of the requirements changes. These reports contain far more technical and historical detail than presentations prepared for the Shuttle software change approval process. The reports were published and archived in the JSC technical library (Fig. 9b).

[34] Goodman, J. L., *STS-49 Lambert Targeting Anomaly and Aftermath*, JSC-49710, Flight Design and Dynamics Division, NASA JSC, May 2003.

[35] Goodman, J. L., *Space Shuttle Lambert Cyclic Guidance*, JSC-49709, Flight Design and Dynamics Division, NASA JSC, May 2003.

[36] Goodman, J. L., *Space Shuttle Lambert Targeting*, JSC-49708, Flight Design and Dynamics Division, NASA JSC, May 2003.

[37] Rea, J. R., and Ives, D. G., *Modification of the Space Shuttle Powered Explicit Guidance Code for the X-38 Crew Return Vehicle*, Aeroscience and Flight Mechanics Division, JSC-28764, August 1999.

[38] Ives, D. G., *Shuttle Powered Explicit Guidance Miscellaneous Papers*, Aeroscience and Flight Mechanics Division, JSC-28774, November 1999.

[39] Jaggers, R.F., Long, A. D., and McHenry, R. L., *Exoatmospheric Generalized Guidance For Shuttle*, Mission Planning and Analysis Division, IN-73-FM-168, JSC-08664, December 18, 1973.

[40] Long, A. D., and McHenry, R. L., *Shuttle Powered Flight Guidance Equations Development*, Mission Planning and Analysis Division, 84-FM-37, JSC-19995, August, 1984.

[41] Jaggers, R. F., *Shuttle Powered Explicit Guidance (PEG) Algorithm*, Flight Design and Dynamics Division, JSC-26122, November 1992.

[42] Uzzell, B. R., *Elliptic Lambert For Space Shuttle Onboard Software*, 79-FM-17 Rev. 1, JSC-14905, Mission Planning and Analysis Division, NASA JSC, July 1979.

[43] Harpold, J. C., *Analytic Drag Control Entry Guidance System*, Mission Planning and Analysis Division, IN-74-FM-25, JSC-08974, January 21, 1975.

[44] Harpold, J.C., and Graves, C. A., *Shuttle Entry Guidance*, Mission Planning and Analysis Division, IN-79-FM-7, JSC-14694, February 28, 1979.

[45] Spencer, J. L., *Use of a Nonsingular Potential*, JSC Internal Note No. 75-FM-29, JSC-09661, Mission Planning and Analysis Division, NASA JSC, May 27, 1975.

[46] Bean, W.C., and Price, R. A., *Babb-Mueller Atmospheric Density Model - Calibration And Interface With The Shuttle Onorbit Navigation Flight Software*, Mission Planning and Analysis Division, IN-81-FM-59, JSC-16931, February 28, 1982.

Due to the mass retirement of a number of critical personnel with knowledge of Shuttle sequencing software, the Shuttle Program directed that a system history and rationale document be created.[50] The document is periodically updated to reflect changes in the software requirements. During a project to create a common software library for Shuttle mission design and planning, an engineering manual was created, which presented the theoretical basis for the software requirements and tied the equations to the software architecture.

In 1996, when test flights of the Shuttle's GPS receiver began, one concerned engineer created a four-page document covering GPS parameters for Mission Control personnel. Over five years the GPS Operating Characteristics document grew to 174 pages. It contained extensive technical information on GPS receiver operation and performance, Shuttle computer software design, GPS receiver and Shuttle software change history, flight test results, and the resolution of performance issues. The information captured was not available in formal requirements documents, and was a valuable resource to both NASA and contractor personnel supporting the Johnson and Kennedy Space Centers.

3. Preserving and Improving Presentations and Technical Reports

Presentation charts for meetings are often the only record of an analysis or issue that was discussed. Such charts are not always effective at preserving the analysis, results, conclusions, and decisions made based on the discussion of the charts. Meeting charts are often created for a limited audience, where it is assumed that meeting attendees already have an understanding of why the issue is important and what is motivating the analysis. Charts are often not understandable without an oral discussion by the author. Chart authors may not effectively communicate technical detail, rationale, and assumptions to the audience and may not create the charts so that a future reader will be able to understand the significance of the analysis. A lack of published and appropriately archived meeting minutes compounds the problem.

PowerPoint™ presentations created for the STS-114 Design Certification Review used hyperlinks to include supporting information and documentation, such as test data, video clips, requirements, anomaly investigation reports, process documentation, calculations, and email (Fig. 10). This permitted viewers to access any level of detail that was required thereby ensuring the appropriate level of technical communication occurred during the review. The presentations and hyperlinked documentation and media were contained in shared drives that were a part of the EKB (Fig. 8), thus ensuring preservation and future access.

The quality of communication with presentation charts has long been a concern, particularly in the wake of the *Challenger* and *Columbia* accidents.[51,52] Some concerned management and technical personnel have begun to address the issue of presentation quality by using some of the many resources published to improve technical communication.[53-55] The report of the Columbia Accident Investigation Board in particular has highlighted the limitations of presentation charts as compared to technical reports for communication and knowledge preservation.[52] Management and technical personnel should discern when a detailed, comprehensive technical report should be written, published, and archived. Engineers concerned about knowledge preservation mentor other engineers so that presentations, memos, and technical reports contain useful information, not just data, that will assist future engineers in understanding what was done, and why.

[47] Goodman, J. L., *Improvement Of Space Shuttle Time To Node Computation*, Flight Design and Dynamics Division, JSC-49766, July 28, 2003.

[48] Brownd, J. E., *Space Shuttle RNP Matrix Computation (CR 92329E)*, Flight Design and Dynamics Division, JSC-49834, October 2003.

[49] Meissen, T., *Space Shuttle Lambert Guidance Improvement*, SCR 92843, Flight Design and Dynamics Division, JSC-49830, in publication.

[50] *Guidance, Navigation and Control Sequencing Flight Software Historical Rationale Document OI-30*, NS03HOU138, The Boeing Company, May 30, 2003.

[51] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, U.S. Government Printing Office, Washington, DC, June 6, 1986.

[52] *Columbia Accident Investigation Board Report*, Volume I, U.S. Government Printing Office, Washington, DC, August 2003.

[53] Tufte, E. R., *The Cognitive Style of PowerPoint*, Graphics Press LLC, Cheshire, CT, 2003.

[54] Mignot, J., "Helping Engineers and Scientists Avoid PowerPoint Phluff," 2005 IEEE Aerospace Conference, IEEE, New York, NY, 2005.

[55] Doumont, J., "The Cognitive Style of PowerPoint: Slides Are Not All Evil," Technical Communication, Vol. 52, No. 1, February 2005, pp. 64-70.

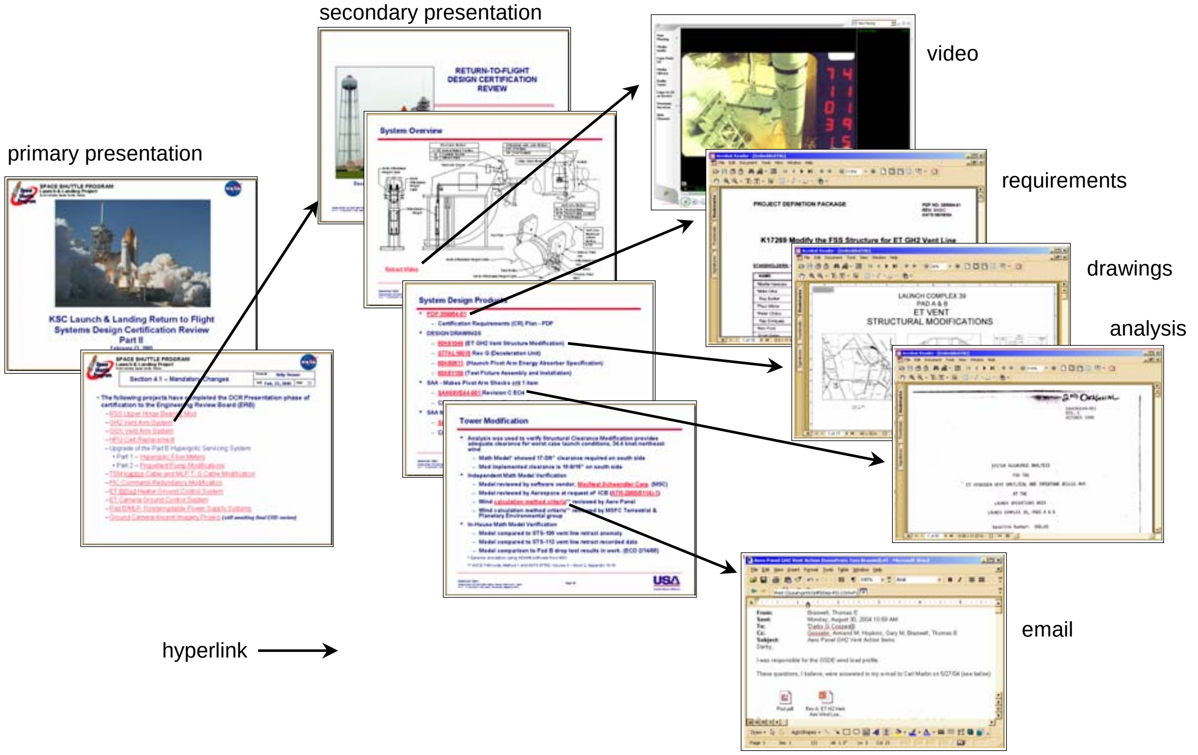


Figure 10. STS-114 Design Certification Review Charts Contained Hyperlinks To Supporting Presentations, Documents And Other Media, and Were Stored In The EKB

Many organizations in the Shuttle and ISS programs archive presentations on websites, along with meeting minutes. The minutes help provide the historical context for the presentation material and provide a record of the outcome of the meeting.

In 2003, an engineer conducted an independent review of equation derivations and analysis in support of new requirements formulation for Mission Control GPS filtering software.[56] The technical report contained a complete derivation of the equations in the requirements document. Enough steps and detail was provided to enable new engineers to duplicate the derivations and understand the theoretical origin of the filter requirements.

[56] Goodman, J. L., "Application of GPS Navigation to Space Flight," 2005 IEEE Aerospace Conference, IEEE, New York, NY, 2005.

4. Embedding Rationale in Requirements Documents

Some experienced engineers and software specialists embed supporting rationale for particular requirements within software requirements documents. Embedding rationale within the flight rules documents has greatly aided Mission Control personnel years later when proposing changes or evaluating the effectivity of the flight rules.

5. Web-Based Knowledge Capture and Management During Software Development

In 2004, a group of Mission Control software developers began using an online knowledge base encyclopedia that permits users to quickly create, modify and link web pages (Fig. 11). This "one stop shopping" application provides easy access to frequently referenced and dynamic information; such as software schedule data, software build versions, development procedures and Shuttle mission specific information. In addition, the database is also used as a repository for solutions to problems and infrequently referenced information. It has also facilitated more convenient capture and preservation of information by departing employees. Database simplicity and ease of use

Figure 11. Some Mission Control Software Developers Are Using a Web-Based Application to Capture and Manage Knowledge Not Normally Contained in Formal Software Documentation

has aided integration into daily work activities, and has improved overall knowledge retention and access.

Managing Talent and Changing Culture

Over the long term, proper documentation and preservation of critical knowledge should become a normal and expected part of engineering tasks (what was done, why it was done, how it was done, how results of the work were implemented or factored into decision making). Management could identify those engineers who possess writing skills, and assign them to tasks that are poorly documented. For example, an engineer with writing ability can be assigned to work with a talented theoretician who may not take the time to document his or her work. This approach takes advantage of the complementary talents that engineers in an organization may possess.

Engineers who enjoy writing, and possess good presentation skills, can mentor those engineers who need to develop skills in those areas. Some organizations offer in-house writing and presentation training and books for self-study. Training young engineers to properly present, document and preserve their work is necessary to ensure the continuance of appropriate knowledge preservation habits throughout their careers.

Engineers should be curious about the theory used in software tools, the limitations of the theory in the application in question and on-board and ground systems performance, history, and design rationale. A desire for learning and investigation is a necessary component of any program that uses complex technology in safety and mission critical applications. Organizations that promote and value intellectual curiosity, theoretical understanding, and performance investigation may be more effective at retaining talented engineers.

Improving Knowledge Capture and Management In Future Programs

A new vehicle program would present optimal opportunities to build knowledge capture and management into a program from the beginning. Knowledge capture and management must be embodied in the new program throughout its entire life cycle. This is essential if safe, predictable, and sustainable performance of flight vehicles and ground systems are to be maintained through the life of the program. Accurate and complete information from today provides a basis for decisions that may be made many years in the future and long after the people who generated that information have gone (Fig. 12).

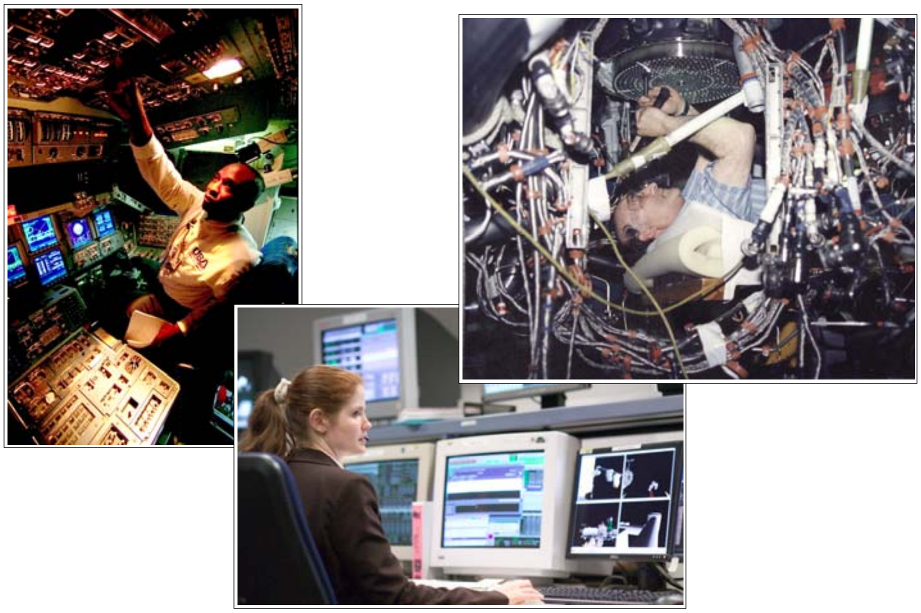


Figure 12. Today's Work May Be Tomorrow's Solution – If It Can Be Remembered

To effectively address the problem, knowledge capture and management software should reside in a program wide process governed by clear, concise, and non-negotiable requirements. These requirements should reflect the value that knowledge capture and management has for reducing risk and lowering life cycle costs. Program wide knowledge capture and management processes and mechanisms should be considered an integral part of the safety and quality control program and an important part of a safety culture.

Well-written requirements provide flexibility in what methods are used to perform knowledge capture and management, while also ensuring that the knowledge activities are conducted consistently across a program and provide benefits over the long term (Table 2). The requirements should ensure that the same rigor that is applied to document new technology and lessons learned early in the development phase is maintained later in the program. Goals are identified, along with the types of problems the policy is designed to mitigate over the life of the program. Guidelines assist management and technical personnel in defining what historical engineering information for fight vehicle and ground systems, facilities, and equipment must be captured and managed, as not all knowledge that management and technical personnel possess can be captured (Table 3). Policies and guidelines also provide direction in various areas such as the extent of deliverable content, preservation of knowledge during the transition from the development phase to the flight phase, and the recording and dissemination of technical and programmatic lessons learned.[57] Accountability and the provision for audits to ensure conformance with requirements should also be addressed. Provisions should be made to develop capture processes later in the program, if it is recognized that some knowledge is not captured through formal means. Identification of current and former subject matter experts should also be addressed in the requirements.

[57] Rogers, E. W., and Milam, J., "Pausing for Learning: Applying the After Action Review Process at the NASA Goddard Space Flight Center," 2005 IEEE Aerospace Conference, IEEE, New York, NY, 2005.

Table 2. Knowledge Capture and Management Options and Benefits

	Options	Actions	Benefits	
Existing Documents	A. Discovering What Documents Presently Exist	Create Bibliography of Critical Surviving Documents Ensure Local Preservation Identify and Address Poorly Documented Subjects	Material is identified that contains important information. Engineers will know what historical material is available.	Reduces or eliminates the need to conduct searches for documents that may or may not have survived. Engineers may be able to access the material without encountering “roadblocks.”
	B. Preserving Existing Documents	Collect and Centrally Archive Surviving Documents in a Repository	Long term preservation of surviving material is ensured.	Engineers can access surviving material without difficulty.
Future Documents	C. Modifying Processes to Ensure Knowledge Capture	Adjust Standards For “Deliverables” Embed Within Processes the Creation and Review of Knowledge Capture Documents For Selected Work	Processes ensure that critical knowledge and rationale will be captured. Impresses on engineers the importance of proper documentation of theory, analysis and requirements rationale.	More community members become familiar with theoretical concepts and requirements rationale, enhancing the quality of analysis, software changes and safety.
	D. Companions to Requirements Documents	Place Under Configuration Control Contains Requirements Rationale and Theoretical Development	Complete documentation of theory and requirements rationale. Critical knowledge retained in spite of employee attrition or contract changes. Makes the process of researching theory and requirements rationale less time consuming.	Eliminates need for engineers to “reverse engineer” to discover underlying theory and requirements rationale. Less experienced engineers not as dependent on mentoring by veteran engineers.

Table 3. Summary of Material to be Captured and Archived (Not Exhaustive)

Concept Studies	Critical Item List Acceptance Rationale	Presentations
Trade Studies	Hazard Mitigation	Engineering Notes
Analyses of Design	Failure Modes And Effects Analysis	Preliminary Design Reviews
Operational Alternatives	Post Flight Reports	Critical Design Reviews
Rationale For Design Selection	Ground and On-board Procedures and Supporting Analysis and Rationale	Meeting Minutes
Engineering Analyses	Life Cycle, Maintainability, and Other Design Considerations	Operational Readiness Reviews
Drawings and Supporting Calculations		Program Board Activities
Physical and Mathematical Models	Rationale For Systems Design And Component Selection, Qualification and Certification	Derivations of Equations
Risk Assessments		
Launch Commit Criteria and Supporting Rationale	Ground and On-board Instrumentation Requirements And Data	
Maintenance Requirements and Specification Supporting Rationale	Test and Validation Requirements and Data, Systems Performance Data	
Flight Rules and Supporting Analysis and Rationale	Vendor and Supplier History And Issues	

Experience from past programs is useful in pinpointing which processes need special knowledge capture and management mechanisms built into them (Fig. 13). Valuable input concerning knowledge capture and management

Figure 13. Lessons and Best Practices From Legacy Programs Are The Key To Improving Knowledge Capture and Management In The Future

requirements can be obtained from engineering and management personnel who faced the problem on legacy programs. Such personnel can help define how deliverables can be tailored to perform knowledge capture, and what knowledge needs to be captured that is not normally contained in formal or informal documentation. Deliverables that are internal to an organization and not contractually required by the customer may have to be created to ensure knowledge preservation.

Requirements should be established governing appropriate repository systems for archival knowledge. This would include the consolidation and/or cataloging of existing systems and both formal (published within an organization or program) and informal (hardcopy or electronic media from a personal archive) knowledge. A desirable objective is to minimize the number of archives where knowledge is located and to avoid potential duplication of information. A search and retrieval capability that provides easy access and the ability for single searches across multiple information sources (a single "on-ramp" to "one-stop shopping") is desirable.

Knowledge capture mechanisms should be embedded in specific engineering processes, especially those that routinely handle important technical, risk, and safety issues such as requirements, software and hardware development, testing, and technical and program level boards. This is needed to acquire engineering knowledge as it occurs while preserving the original format and intent. Collecting important information in this manner is a more cost effective and robust approach than trying to create new documentation, or capture surviving documents years after key personnel have left the program.

The ideal knowledge capture document for software would be a configuration controlled, companion document to the software requirements, that contains a complete disclosure of requirements rationale and equation development of algorithms implemented in the software. All simplifying assumptions and mathematical identities and manipulations would be recorded to enable future engineers to duplicate the results. If equations developed in open literature resources are used, any differences in notation should be recorded. Other publications, either in the internal or open literature may be referenced in the companion document. However, internal materials referenced must be archived in a manner that facilitates easy location and access over time. The companion document would also tie the equations to the software architecture. Such a reference would enable new engineers to become educated more quickly, and eliminate the need to perform time consuming and costly re-engineering. The Goddard Trajectory Determination System Mathematical Theory document is an excellent example.[58]

[58] *Goddard Trajectory Determination System (GTDS) Mathematical Theory*, Revision 1, FDD/552-89/001, CSC/TR-89/6001, Goddard Spaceflight Center, Flight Dynamics Division (Code 550), July 1989.

Updates would be made to the companion document as changes were made to the software over the life of the flight program. Review of the equation document would be embedded in the existing requirements and software process. The proposed changes to the equations document would be distributed in time to support software change testing, and would be reviewed by engineers and software specialists for technical accuracy, completeness, clarity, and compliance with documentation requirements. Independent reviews of updates to equation derivations and analysis contained in the companion document would ensure a robust software maintenance process and the honing of technical skills. Final approval of software change requests would be contingent upon approval of the update to the companion document by engineering and software personnel.

COTS applications will undoubtedly form the basis of future knowledge capture and management efforts. COTS products should be usable and maintainable over the life of the program. Critical information and archival capabilities should not be lost due to changes in host computer platforms, changes in operating systems, or COTS product obsolescence. Archival media (such as paper or electronic) should not deteriorate over the life of the program, so that it will remain intact and accessible for posterity. Knowledge capture and management requirements should address these concerns.

Conclusion

As time progresses, corporate knowledge loss within the space flight industry will become more of a challenge. Knowledge capture and management is not a technical issue, but a cultural one. Underlying cultural and programmatic issues that prevent knowledge from being captured and managed must be addressed before information technology can be leveraged to address the problem. Leadership from both management and technical personnel is needed to foster and sustain a culture where intellectual curiosity, effective communication, and the creation of proper documentation are valued. Personnel and processes can be managed so that proper documentation of critical knowledge concerning both hardware and software is performed and becomes a part of the program culture. It is no longer adequate for engineers and managers to address this on an individual basis as has been done in the past, but it should be elevated to the programmatic level to facilitate process and cultural change. Management and technical personnel can assess the state of knowledge capture within their organizations and devise creative, low cost ways to address the issue (such as the examples given on pages 13-19). To avoid knowledge capture issues in the future, documentation requirements and knowledge capture mechanisms should be built into analysis, software, and hardware processes at the start of every new program. Throughout the life of the program, it will be necessary to capture knowledge gained from testing, certification, procedures development, and continuous use (such as the effects of wear, degradation, and obsolescence). It is essential that the underlying information and knowledge associated with problem resolution be captured, not only for the purpose of avoiding "re-inventing the wheel" (avoiding the 2nd "first time"), but also for the identification of systemic issues and taking pre-emptive action to predict and mitigate potential failures.

Acknowledgements

Jeffrey F. White, with United Space Alliance at the Kennedy Space Center, provided helpful information on knowledge capture and management from the ground systems and facilities perspective. Mr. White was the driving force behind creation of the Engineering Knowledge Base at the Kennedy Space Center, and has been working on the knowledge capture and management issue for over twenty years.