Risk Management for the B-1B Computer Upgrade

2nd Lt Daniel P. Stormont
B-1B System Program Office, Aeronautical Systems Center
Capt Robert Welgan
Air Force Institute of Technology

ABSTRACT

The B-1B has received a great deal of criticism since it was first fielded in 1985. Much of this criticism, deserved or not, was the result of perceived shortcomings in the avionics system--especially the Defensive Avionics System. These shortcomings were largely the result of software deficiencies. The B-1B is now entering a conventional mission upgrade, which will include replacing some of the computers and modifying some of the software. The challenge for the B-1B Conventional Weapons Team is to manage the risk inherent in this computer upgrade in order to prevent more criticism in the future. This will require a new approach to the way risk has been managed for the B-1B.

This paper briefly describes the history of the B-1B's computer hardware and software, risk management techniques used previously on the B-1B program, lessons learned, and risk management plans for the conventional mission upgrade now under way for the B-1B.

INTRODUCTION

Ever since it's inception as the B-1A bomber, the B-1 has been the subject of a large amount of negative press coverage[1]. When President Ronald Reagan made the decision to produce the B-1 as the low-level penetrator B-1B variant in October 1981[2], the barrage of negative coverage intensified[1][3]. Much of this coverage centered on the early mechanical problems, such as fuel leaks and asynchronous wing sweeps. The crash of aircraft 84-0052 due to a bird strike and 85-0063 due to an on-board fire intensified the focus on mechanical problems. However, the majority of negative press coverage has been reserved for avionics problems, especially concerning the perceived inadequacy of the AN/ALQ-161A Defensive Avionics System[4][5]. The shortcomings of the B-1B avionics systems have largely been the result of software problems[5][6][7].

In June 1992, the Department of the Air Force released The Bomber Roadmap. This plan called for new conventional roles for the primary nuclear deterence bombers, the B-1B and the B-2A. In order to achieve the capabilities required by the roadmap, the B-1B will require a systematic series of upgrades over the next fourteen years[8]. Nearly all of these upgrades will require the development of additional weapon delivery software. They will also require the replacement of some of the current computers on the B-1B, which are rapidly reaching the limits of their design capacity[9].

This is the challenge that this paper attempts to address--managing the upgrade of the B-1B's computers and associated software, without adding to the negative reputation that this aircraft has so unjustly earned. This requires the application of judicious risk management procedures in all phases of the upgrade program. This paper will examine these procedures in depth: beginning with a definition of risk management, reviewing the history of the B-1B's computer hardware and software development, and finishing by presenting the current actions of the B-1B upgrade team to identify and manage the risks associated with software development and hardware upgrades.

I. RISK

Risk is the probability that some unacceptable outcome, resulting in a loss, will occur. There are three broad categories of risk: cost, schedule, and technical. Cost risk is the chance that the program will exceed its budget. Schedule risk is the chance that the program will exceed it's schedule. Technical risk is the chance that the program will not meet its technical objectives. A successful risk management program consists of two parts. First, risk must be assessed. Second, risk must be placed under some control. Members of the B-1B Conventional Weapons Team have employed informal methods of managing risk for the B-1B computer upgrade. A thorough understanding of methods to assess and control risk, and an understanding of the current risk management technique, may lead to improvements in risk management for the B-1B computer upgrade.

The risk management task can be decomposed into two subtasks: risk assessment and risk control. These may in turn be decomposed further (Fig 1)[10]. The first step in risk assessment is risk identification. There are a number of ways to identify risk. First, one may use a checklist of known risks that have occurred on similar programs and compare them to determine possible future risk to the program. Second, one may perform decision driver analysis to determine whether or not key program decisions have been made for reasons other than technical or management considerations. If so, they may be a source of program risk. On the B-1B program, the primary methods of identifying risk have been assumption analysis and decomposition. Assumption analysis identifies optimistic assumptions made about the program; these are sources of risk. Decomposition involves breaking the design down into small parts and analyzing the difficulty of producing components with the desired capability given the resources allocated. This helps management determine which subset of the proposed components will demand the most attention during the project.

The second task in the risk assessment process is risk analysis. This involves determining what the program's identified risks translate to in the way of cost overruns, schedule delays, and technical shortfalls. For the B-1B computer upgrade, the cost models REVIC, SEER, and PRICE-S have been used to assess the cost and schedule risk. Performance models and trade studies help characterize the program's technical risk. Network analysis has formed the basis for several studies of B-1B computer requirements by breaking functionality down into network nodes and identifying possible combinations of these nodes as potential architecture structures.

Once program risks are understood, they are placed in priority order, so management knows how to allocate resources against them. This prioritization is based on the concept of risk exposure. Risk exposure values for each risk item can be calculated using a weighted cost value determined by the following equation:

$$RE = P(UO) * L(UO)$$

where P(UO) is the probability of unacceptable outcome and L(UO) is the loss, usually in dollars, due to an unacceptable outcome. These values can be used to determine the benefit of risk reduction in comparison to the costs of the risk abatement effort required. The results of this cost-benefit analysis can be used to rank order the risks according to risk leverage[11]. A weakness of the B-1B program has been a lack of this risk leverage analysis. Risk prioritization has typically been based on "expert opinion" within the team, due to insufficient data for determining probability and loss values in the risk exposure equation. Improvements here are key to ensuring program resources are properly allocated to the program's risk control effort.

After risks have been assessed, management is ready to take steps to control these risks. The steps of risk management are planning, resolution, and monitoring. Plans should be developed for each risk element. Barry Boehm recommends these plans include the following:

1. Why the risk is important, and why it should be managed.
2. What should be delivered regarding risk management and when.
3. Who is responsible for performing the different risk management activities.
4. The method of risk abatement.
5. The resources needed to reduce the risk[10].

The B-1B System Program Office (SPO) has not yet developed these plans for each risk element associated with the B-1B computer upgrade.

II. B-1B COMPUTER ACQUISITION HISTORY

As the introduction pointed out, the B-1B has had a troubled history. Media and congressional scrutiny has been severe throughout the aircraft's lifetime. This has sometimes caused less than optimal solutions to be adopted for political reasons. The B-1B is a very effective combat aircraft; however, the capabilities of it's current computers are very limited.

The B-1B employs a network of eight International Business Machines (IBM) AP-101F general purpose avionics computers to control and coordinate it's on-board avionics systems (Fig 2)[13]. These computers each have two central processing units (CPUs), one for the MIL-STD-1750A instruction set architecture (ISA) and one for the IBM proprietary multi-purpose midline processor (MMP) ISA. They also have either 128 or 256 kilowords (KW) of semiconductor and magnetic core memory each. The AP-101F is rated at 1 million instructions per second (MIPS). (See Figure 3 for a block diagram of the AP-101F computer[13]).

The software for the B-1B is targeted to the MMP ISA in the AP-101F. This ISA was chosen because of the lack of a certified MIL-STD-1750 compiler at the time the system was fielded and because the B-52 Offensive Avionics System (OAS), which used AP-101C computers, already had software developed for the MMP ISA. The software was never retargeted for the 1750 ISA, as was originally intended, because there was no desire to modify the software once it was working.

The software for the AP-101Fs is divided into the Avionics Offensive Flight Software (AOFS), Preprocessor Flight Software (PFS), and Central Integrated Test System (CITS) software. The majority of the code is in the AOFS, which is the software that controls the on-board avionics. The AOFS is modularized by function (as shown in Figure 4)[13]. The functions are partitioned into load modules for each of the computers.

The B-1B computer system requires programs written in a real-time programming language. Real-time programming languages allow the programmer to strictly control the time it takes for the program to complete its tasks. If these computer functions are not completed at precisely the right time, the B-1B's mission will be a failure. The first B-1B contractor chose the Air Force standard real-time programming language, JOVIAL version J3B2 as the systems programming language.

JOVIAL has several features which at the time were thought to be important for a real-time programming language. First, the language is weakly typed. This means the compiler does not catch errors where references to program variables contain mixed types. Second, it allows direct access to the underlying computer's machine language. It allows the programmer to selectively modify bytes and nibbles in the computer's memory. Once people learn how to use JOVIAL, they find it to be a very powerful language. However, it is difficult to learn and not in widespread usage[12].

With JOVIAL's powerful language features and learning curve, software development can be risky unless a number of steps are taken. First, any people added to the program should be trained extensively. Second, programming standards should be written to ensure special attention is given to in-program comments and control of variable typing. Also, a vigorous program of frequent, informal code inspections by peers, throughout the system's life, will help ensure the language's powerful features are well-used.

In addition to the potential problems inherent in the use of the JOVIAL language on the B-1B, the computers are rapidly reaching the limits of their design capacity. When the B-1B first became operational, the computer system was well below the Air Force specified memory reserve of 30% and the specified throughput reserve of 50%[13]. With the latest release of AOFS, which is being fielded now, spare memory has been reduced to as little as 3.8% and spare throughput to as little as 29.2%[14]. In order to add new functionality to the computer complex, it has been necessary to "roll in" code from the mass storage device because of the memory constraints. And, in critical portions of a typical flight profile throughput utilization can exceed the current capacity of the computers. This causes a "no-go" condition and automatic reconfiguration of the computer (Fig 5)[15]. This can be problematic if it occurs at a critical point in a mission.

III B-1B COMPUTER UPGRADE RISK MANAGEMENT

The best teacher is experience. For this reason, the B-1B Conventional Weapons team has tried to apply the lessons learned during the initial procurement of the B-1B[16]. There were many valuable lessons learned, but some of the lessons most appropriate to the current upgrade are:

"You can never have enough spare memory and thruput."

"Never rely on contractor claims that any firmware item will not require change over system life cycle."

"Conducting a flight test program with a fly/fix/fly strategy is very inefficient."

As one step toward applying past experience to the current upgrade, a Top 12 Risk Item chart was developed for identifying and tracking the status of items likely to cause program risk[17]. The key to an effective Top 12 chart is correctly identifying items that are program risks, ensuring agreement among the team members and management about the risk items, actively tracking the items, and taking action to correct them.

One step toward reducing risk on the B-1B computer upgrade was early software cost and schedule estimating for the planned upgrade effort. Per current Aeronautical Systems Center (ASC) policy, these estimates used three commonly used models. These models were PRICE-S, REVIC, and SEER. The PRICE-S model was run by the ASC/ALT office[18]. The REVIC and SEER models were run by members of the B-1B teams. There was a large variance in the results from the models, even given identical inputs (Fig 6)[19]. This variance was dealt with by "sanity checking" the values against contractor rough order of magnitude (ROM) cost estimates, actual costs from similar efforts, and data from previous B-1B development. These efforts were repeated for hardware cost and schedule estimates using the PRICE-H model.

One of the primary risk reduction efforts is the contracting of independent trade studies on areas with high risk. A study was conducted to clarify the computer requirements and to identify options that will satisfy those requirements. A study also was conducted into the risks and benefits of Very High Speed Integrated Circuit (VHSIC) technology as an upgrade alternative. Another study investigated potential difficulties associated with emulating the AP-101F architecture with off-the-shelf processors. A study that is currently underway is investigating the conversion of JOVIAL J3B2 code to Ada.

An organizational risk has resulted from the reorganization of Air Force Logistics Command (AFLC) and Air Force Systems Command (AFSC) into Air Force Material Command (AFMC). This reorganization has introduced the concept of Integrated Weapons System Management (IWSM). What this essentially means is that the acquisition and support agencies are one in the same. This doesn't have much effect on a new program, but, since the B-1B had already gone through Program Management Responsibility Transfer (PMRT) in the old organizational structure, this means that the B-1B program office is geographically divided: "SPO North" (ASC/SDB at Wright-Patterson AFB, OH) and "SPO South" (OC-ALC/LAB at Tinker AFB, OK). These two agencies must present a "single face" to the user, Air Combat Command (ACC). This requires an incredible amount of communication and coordination, even for seemingly trivial items. This task is complicated even more by the disparity in objectives and motivations between the two organizations. One way this has been dealt with is through the use of scheduled, frequent video conferences between the teams. A common electronic mail directory has been set up. Also, members of both organizations co-chair important formal working groups (e.g., Computer Resources Working Group, Test Planning Working Group) and subgroups.

CONCLUSION

The B-1 program has been the subject of a lot of criticism and Congressional scrutiny throughout it's history. It has been an ambitious program, full of technical, cost, and schedule risk for all of its major subsystems, especially the AN/ALQ-161 Defensive Avionics System. These risks are not reduced by the choice of real-time programming language, computer hardware capacity and capability, geographic division of the program office, and political sensitivity of the program: risks are increased by these characteristics. While the B-1 program has had success with certain risk management techniques, the true effect of these techniques has not been completely quantified, due in part to a history of informal risk management practices. Whether the failures of the B-1 program can all be attributed to its current informal risk management process is unclear.

One thing is clear: the program office will continue to be challenged by cost, schedule, and technical risks during the upgrade of the B-1B. If there is to be any hope of avoiding problems with the new computer hardware and software, the risks inherent to their development must be taken seriously. The program office already is taking action by concentrating efforts on the top twelve risk items. However, this alone is not enough. Specific plans to identify, prioritize, analyze, and control risk must be written. Then resources must be allocated to mitigate the risks and assess the success of these mitigation efforts. These plans will permit mangament to exercise needed control over the program's risk management effort. This way, not only will the B-1 program be able to take advantage of it's lessons learned, it can redirect its risk reduction efforts to match fluctuations of program requirements.

REFERENCES

1. T.R. Milton, "Through Flaps and Flak", Air Force Magazine, January 1989, pp. 80-84.
2. W. Thompson, "Bombing in the B-1B", Airpower, January 1989, pp. 26-46.
3. F. Oliveri, "The Bone", Air Force Magazine, September 1992, pp. 34-40.
4. D.M. North, "Development Problems Delay Full B-1B Operational Capability", Aviation Week & Space Technology, 3 November 1986, pp. 34-35.
5. "Defensive Systems to Undergo Full Flight Test in 1988", Aviation Week & Space Technology, 14 September 1987, pp. 69-77.
6. R.S. Dudney, "Strategic Modernization in the Shakedown", Air Force Magazine, October 1990, pp. 44-46.
7. J. Rhea, "The Next Generation of Avionics", Air Force Magazine, January 1990, pp. 68-72.
8. The Bomber Roadmap, Department of the Air Force, June 1992.
9. B-1B OAS Data System Performance Margins and Limitations, Boeing Military Airplanes, Seattle, WA, 17 September 1990.
10. P. Jalote, An Integrated Approach to Software Engineering, Springer-Verlag, New York, NY, 1991.
11. B.W. Boehm, "Software Risk Management: Principles and Practices", IEEE Software, January 1991, pp. 32-41.
12. R.L. Wexelblat, History of Programming Languages, Academic Press, New York, NY, 1981.
13. B-1B OAS Maintenance Training Course, Week 1, Boeing Military Airplanes, Seattle, WA, 15 October 1985.
14. T. Lam, AOFS Block 4.7 CDR Software Performance Measurements, Oklahoma City Air Logistics Center, Tinker AFB, OK, 21 July 1992.
15. B-1B Offensive Avionics Systems Mission Critical Computer Resource Study, Logicon, San Pedro, CA, 28 August 1990.
16. J. Madia & J. Joseph, B-1B Software Lessons Learned, ASD/B1, Wright-Patterson AFB, OH.
17. B-1B Computer Upgrade Top 12 Risk Items, ASC/SDBW, Wright-Patterson AFB, OH, 5 October 1992.
18. B-1 Computer Cost Estimate, ASC/ALT, Wright-Patterson AFB, OH, 12 August 1992.
19. D. Stormont, B-1B Computer Upgrade Options and Analysis, ASC/SDBW, Wright-Patterson AFB, OH.

AUTOBIOGRAPHICAL SKETCHES:

2nd Lt Daniel P. Stormont is currently the project manager for the B-1B computer upgrade, B-1B Conventional Weapons Team, Aeronautical Systems Center, Wright-Patterson Air Force Base, Ohio. He received his BS in Computer Engineering from the University of Arizona in 1991. He is a part-time student in the Graduate Computer Engineering program at AFIT. Lt Stormont enlisted in the United States Air Force in 1980 as an Electronic Warfare Systems technician. After assignments working on F-4E, F-16A/B, and B-1B aircraft, he received his commission through the Airman Education and Commissioning Program. He is a member of the IEEE Computer Society and the Association for Computing Machinery.

Capt Robert Welgan is a graduate student at AFIT. He has experience with space-based software development and simulation programs.

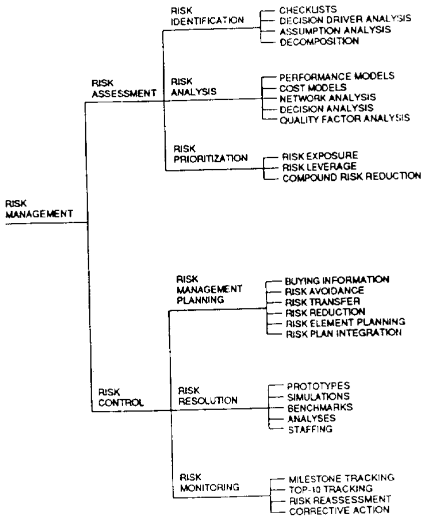
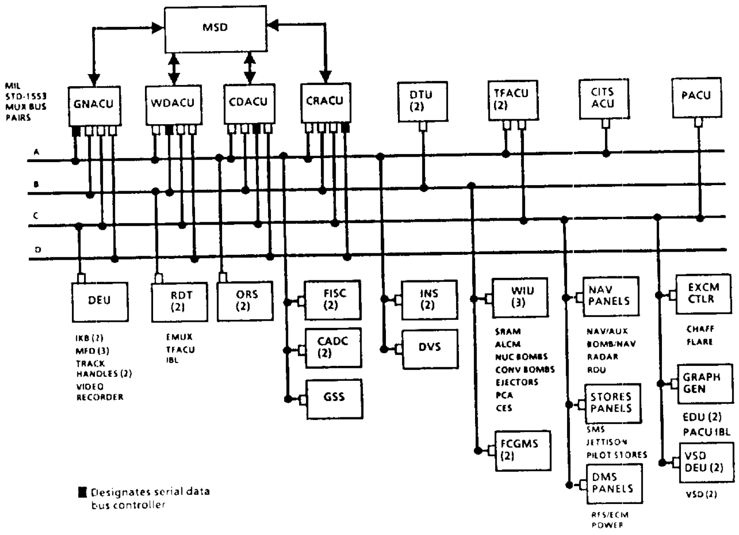


Figure 1



B-1B Computational System Architecture

Figure 2

AP-101F Avionics Computer-Control

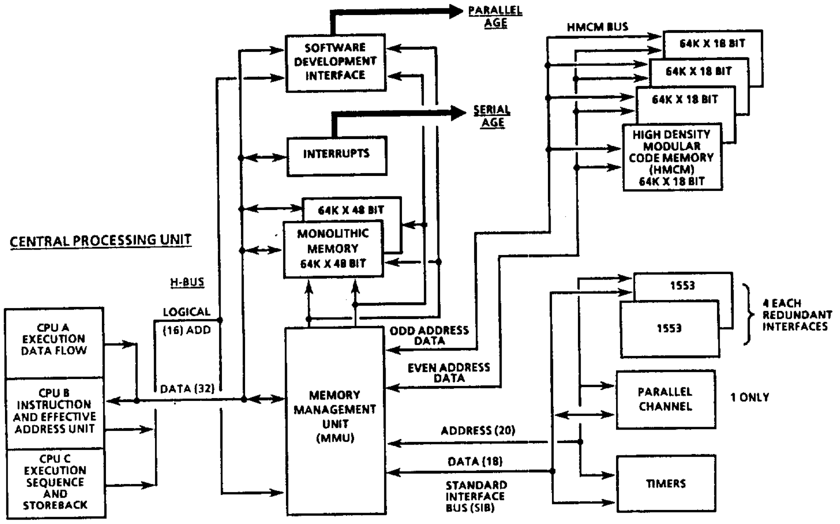

PARALLEL AGE
HMCM BUS
SOFTWARE DEVELOPMENT INTERFACE
64K X 18 BIT
64K X 18 BIT
64K X 18 BIT
SERIAL AGE
HIGH DENSITY MODULAR CODE MEMORY (HMCM) 64K X 18 BIT
INTERRUPTS
64K X 48 BIT
MONOLITHIC MEMORY 64K X 48 BIT
CENTRAL PROCESSING UNIT
H-BUS
LOGICAL
(16) ADD
1553
1553
4 EACH REDUNDANT INTERFACES
CPU A EXECUTION DATA FLOW
ODD ADDRESS DATA
EVEN ADDRESS DATA
CPU B INSTRUCTION AND EFFECTIVE ADDRESS UNIT
DATA (32)
MEMORY MANAGEMENT UNIT (MMU)
PARALLEL CHANNEL
1 ONLY
ADDRESS (20)
CPU C EXECUTION SEQUENCE AND STOREBACK
DATA (18)
STANDARD INTERFACE BUS (SIB)
TIMERS

Figure 3

Avionics Operational Software

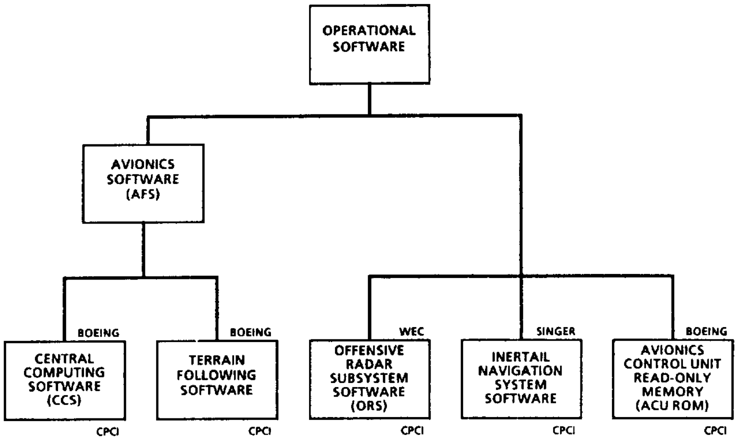

OPERATIONAL SOFTWARE
AVIONICS SOFTWARE (AFS)
BOEING
CENTRAL COMPUTING SOFTWARE (CCS)
CPCI
BOEING
TERRAIN FOLLOWING SOFTWARE
CPCI
WEC
OFFENSIVE RADAR SUBSYSTEM SOFTWARE (ORS)
CPCI
SINGER
INERTAIL NAVIGATION SYSTEM SOFTWARE
CPCI
BOEING
AVIONICS CONTROL UNIT READ-ONLY MEMORY (ACU ROM)
CPCI

Figure 4

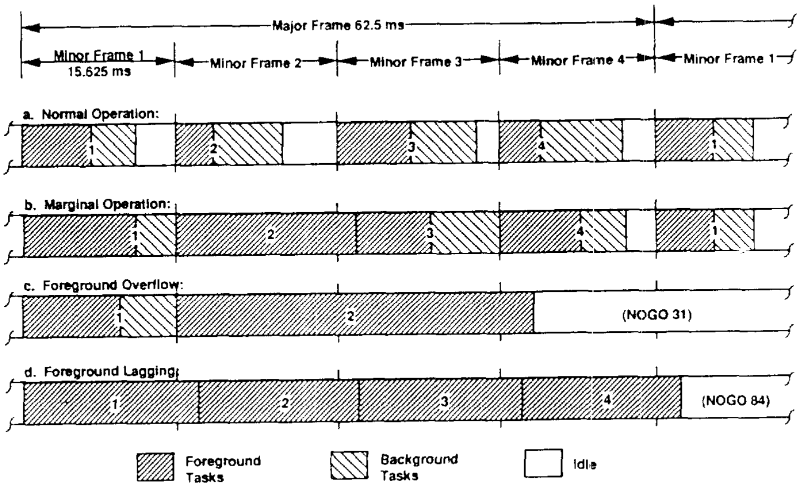


Figure 5

B-1B CWP Computer Upgrade

Software Cost Estimates

Option	Description	Model	Development Cost	15 Year Maintenance	Total Cost
1	Retain J3B2	PRICE-S	$40.8M		
		REVIC	$38.3M	$55.0M	$93.3M
		SEER	$25.5M	$208.1M	$233.6M
2	Ada	PRICE-S	$93.7M		
		REVIC	$23.3M	$33.4M	$56.7M
		SEER	$43.1M	$182.0M	$225.1M

(Assumes an experienced contractor)

Figure 6