

Cost & Schedule – The Overlooked Hazards

Ron Stroup; Certification and Safety Lead
Federal Aviation Administration;
Washington, DC

Warren P. Naylor; System Safety Manager;
BAE SYSTEMS
Rockville, MD

Keywords: Cost overruns, Schedule overruns, Causal factors, Residual risk

Abstract

Absolutely no one intentionally builds an unsafe system! However, systems are routinely built that are not as safe as they reasonably should be. Some of these systems are built by qualified systems engineers, professional safety professionals, and are managed by program managers, which employ the latest software and development methodologies, yet the end product routinely misses expectations. How does this happen?

Cost and schedule overruns are often overlooked as a primary causal factor as to why these systems failed to achieve desired performance and residual risk objectives. With the advent of Acquisition Reform and the impetus for building and deploying systems Better, Faster, and Cheaper, it appears that the better is often overshadowed by the faster and cheaper.

Schedules for programs have become increasingly more aggressive, contracts have become increasingly more restrictive, and start dates are continually pushed back without corresponding relief on the back end, resulting in extremely compressed schedules. Schedule overruns and their accompanying cost overruns have become the rule rather than the exception. Failure to recognize and address cost and schedule as causal factors could result in an avoidable catastrophic event.

Introduction

Our (authors) experience bridges both the civilian and military domains and has

encompassed every avenue from engineering, management, test, certification, and field support. During the span of our experiences we have seen many trends, process improvements, new developmental tools, improvements in the way we design and build systems and the ever present budgetary and schedule shortages and overages.

The intent of this paper is not to assign blame, it is to assist the development community in developing safer and ultimately better products by identifying a deficiency that we believe most recognize but feel powerless to correct. The purpose of this paper is to identify the shortcomings in this demanding environment while providing assistance in mitigating them. No silver bullet(s) will be presented, however, there are ways to reduce the likelihood and potential consequences of systems under severe cost and schedule stress.

The one constant we have observed over our careers is that shortcuts are taken when budgets and schedules become tight. Decisions to mitigate cost and schedule overages are usually comprised of:

- Reductions in developmental testing
- Reductions in integration testing
- Shortcuts on standard development processes (e.g. reviews)
- Reduction in system functionality
- Reduction in training

Case History #1 – V22 Osprey Program

Our first example is one that has been on the front pages of almost every newspaper and even the lead story of the nightly news. This program

is the V22 Osprey Program. Again, we are not trying to assign blame at the V22 program, its managers, engineers, or the brave members of the armed forces that support its development. Our purpose is singular; learning the lessons of the V22 could save future programs from suffering the same fate.

The following quote was taken directly from the front page of the Washington Post. “To save time and money, ... omitted tests of the V22 Osprey that would have provided additional data on rapid descents that contributed to a crash that killed 19 Marines in April, according to a new report by the General Accounting Office (ref. ¹).”



Figure 1 - V-22 Osprey

Unfortunately, as evidenced with the Osprey program, systems testing falls at the end of the development process regardless of the development model used, thereby, becoming a casualty of schedule and cost overruns. Unfortunately, the Osprey Program was so over budget and schedule and under severe scrutiny by both the media and Congress, that the maintenance and flight availability data were allegedly falsified to ensure the program's ultimate survival.

The Osprey Program was and remains in severe jeopardy of being canceled and the V22 capabilities are highly regarded and needed by our combat forces. The Program was under constant scrutiny, review, and pressure forcing tough decisions under extremely stressful conditions. How could errors have been avoided under these kinds of conditions?

Case History #2 – Advanced Automation System

The Federal Aviation Administration's Advanced Automation System (AAS) was a challenging program to replace the computer hardware and software, including controller workstations, in en-route, terminal, and tower air traffic control facilities. Also, AAS was intended to provide new automated capabilities to accommodate increases in air traffic.

The AAS software was ranked among the most complex software development projects in the world and was expected to operate in a real-time environment in which hundreds of functions must be executed within seconds and was expected to be fault tolerant.

The shot gun approach to developing a system of the complexity of AAS proved ineffective. The program failed to meet their defined objectives; specifically (ref. ²):

- Failed to meet reliability objectives
- Design contained unwanted features
- Current state of technology could not support the design
- Failure to achieve defined testing objectives
- System as a whole was never deployed.

After experiencing substantial cost and schedule problems, the FAA restructured the program into more manageable pieces. The U.S. Department of Transportation, Office of Inspector General stated in a report. “AAS failed because of over ambitious plans by both the FAA and the contractor, poor FAA oversight of contractor performance in developing software, and FAA's indecisiveness about requirements (ref. ³).”

Case History #3 – X-33 VentureStar

The goal of the X-33 VentureStar was a single stage wedge-shaped vehicle that could point the way to the opening of the space frontier on behalf of everybody – manufacturing, communication satellites by the score, paradigm busting research, a golden age of interplanetary exploration and even adventure tourism (ref. ⁴).

Former Astronaut Carl Mead acknowledged in the Washington Post article “from the outside the project looked like all bad news, but it felt normal.”

The VentureStar’s demise resulted from a number of factors, such as:

- Conflicting requirements imposed by scientific, political, military, and commercial interests.
- Design leapt ahead of economic and technical realities.

The X-33 VentureStar provided a tremendous amount of knowledge. However, it is clear that the technology has not advanced to the point where the development of a new launch vehicle would improve safety and be economically viable.

Mitigation Strategies

There are many strategies that could be listed in this section. Those strategies that can help alleviate cost and schedule overruns are:

- Consistent goals among the stakeholders
- Coordination among stakeholders.
- Proper contract application
- Evolutionary life cycle

Consistent Goals among Stakeholders: In many organizations the role of safety engineering is separate from the development team. This separation can often result in inconsistent goals. As stated earlier in this paper, the safety engineer ensures the safe and effective development and deployment of a system.

The goal of the program manager is to fulfill the requirements pertaining to the design, development, production and delivery of the system in an effective, efficient, and timely manner. In many instances, cost and schedule become the driving factor in meeting the goal (ref. ⁵).

The goal of any project is to achieve a balance in terms of cost and safety. Experience shows that the cost of implementing safety early in the

design is far less than the cost incurred in rectifying of problems later (ref. ⁶).

Management’s attempts to meet strict schedules by reducing functionality and safety constraints, although well intentioned, are often misguided, as the PM is often inadvertently and sometimes advertently not informed of the inherent risks these shortcuts induce on the project. Over time the safety margin is eroded and there are no up-to-date data on the current or proposed residual risk of the deleted functionality. Programmatic risk is the driver in this situation with safety risk often being overlooked or set aside.

Unfortunately, testing to determine the right balance between safety and cost is performed via live firings or flight-testing. The process is circumvented and system safety is the short-term victim with the end user eventually receiving a system that is not as safe as it could have been.

Coordination among Stakeholders: When the integrated system or candidate system has been identified and the operational concept validated, a coordinating body would be used to ensure that all responsible organizations proceed according to plan and communicate with each other and with stakeholders about program status. The coordinating body should also ensure the necessary evidence of completion is obtained before next steps are undertaken.

The coordinating body would ensure feedback is provided to senior management concerning the status of program completion and any issues, risks or delays identified. Coordinating the activities of a system development leading to operational approval will enable timely and consistent implementation of systems and services within the National Airspace System (NAS).

Proper Contract Application: The type of contract (fixed price, cost plus, incentive fee agreement with cost ceiling, etc.) can play a large role in the effectiveness or ineffectiveness of a program’s ability to ensure goals are met.

The majority of contracts make payments based on time spent and materials used rather than

timeliness and efficiency. Until the payments are directly linked to the completion of deliverables, there is no incentive for the contractor to control costs or use labor efficiently.

Evolutionary Life Cycle: Big-bang development is not an effective means to develop highly complex safety critical systems. The name of the game is “risk reduction,” which means it must be developed to an evolutionary life cycle process. Modernization of complex software-intensive systems must be evolutionary: develop a new system that performs today’s functions, but are unlike current hardware, expandable. Then add new or enhanced functionality. The FAA’s Federal Acquisition Executive stated, “We need to be more risk averse. We’ve learned not to push the boundaries of science (ref. ⁷).”

Safety’s Role

“In order to devise effective ways to prevent accidents, we must first understand what causes them. Determining the cause of an accident is much more complex than is often imagined. Many categories have been suggested: proximate causes, probable causes, root causes, contributing causes, relevant causes, direct causes, indirect causes, significant causes, and so on. (ref. ⁸)” Safety professionals are trained to root out potential hazards, determine their possible causal factors and ensure they are properly mitigated.

The safety professional’s focus is on the engineering process and its relevant engineering products. Our goal is simple: to ensure the safe and effective development and deployment of a system using the Safety Order of Precedence as our guiding light as illustrated in Figure 2.



Figure 2. Safety Order of Precedence

Management’s focus is primarily on assessing, managing, and reporting programmatic risk and the development engineers’ focus is on developing, validating, implementing, and verifying the systems requirements.

For the most part we are successful from a safety perspective, however, occasionally an accident or mishap occurs that should have been prevented. The reported cause of the accident or mishap is usually listed as human error due to a lack of or improper training, an undetected code or design error, an untested function performed improperly under stress, and so on. The truth is that these causal factors are indeed valid, however, were they the true root cause? Or were they just contributing, indirect or significant causal factors. The true root cause can often be overlooked, as short cuts were taken to recover cost and schedule overages.

The safety community plays a significant and sometimes contributory role in the cost and schedule paradigm. Safety’s contribution although well intentioned can impact the cost, schedule and safety both positively and negatively. A balance must be maintained between system safety, system performance, and all other contributory disciplines with cost and schedule as illustrated in Figure 3, particularly in an integrated environment.

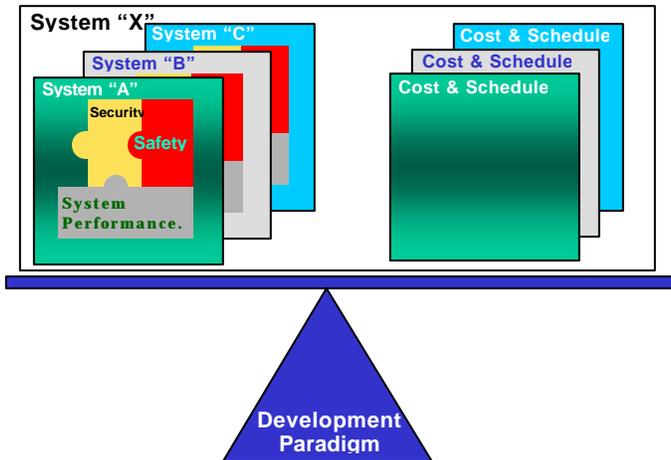


Figure 3. Program Balance

Safety must identify, assess, and report identified hazards as soon as possible in the development process to ensure they are properly and comprehensively mitigated. Failure to do so dooms a system to redesign and rework, resulting in a system that fails to meet its targeted and often even acceptable levels of safety and performance risk.

The standard MIL-STD-882C/D safety analyses are usually the reporting tool of residual risk. Unfortunately these reporting products take time to develop, reproduce, and distribute. The delays introduced by the documentation period are unacceptable and directly contribute to cost and schedule overruns. Each of these analyses is designed to occur at a specific time in a program's development model, regardless of the model chosen, each possessing a specific focus and serving a specific need.

The development products required to perform each of these analyses are time critical and so is Safety's reporting of any perceived design anomalies. Waiting until the design products are complete is unacceptable. Safety must become an active and contributing member of the development team! Anomalies must be acted upon immediately to ensure they are correctly mitigated with little cost and schedule impact. A passive safety program is an ineffective program that will result in cost and schedule overages and a program that falls short of its desired safety margins.

The safety community must also realize that there is a cost associated with employing mitigation techniques. We must propose alternatives that appropriately mitigate the risk while maintaining a balance with cost and schedule concerns. The balance must be maintained between likelihood of occurrence, consequences of occurrence and cost and schedule. There usually is a middle ground that we all (safety and management) can feel comfortable in. Additionally, we have found that we have been much more successful in getting our way when we show management that we also share their concerns regarding programmatic risk.

Please do not misunderstand our argument; there will be times when the safety engineer must recommend a mitigation technique that has a severe adverse impact on programmatic risk. During these times you, as the safety professional, must stand tall and basically put your job on the line. We believe you will find these times will rarely occur if you have developed a reputation for being cooperative, consistent, concerned with cost and schedule and have always presented realistic risk assessments.

Conclusion

The safest systems are systems that design safety in from the start. Retrofitting safety or uncovering safety deficiencies during Test and Evaluation, usually result in systems that are continually asking the question; Have we achieved our target and defined an acceptable level of safety or residual risk.

The impact of cost and schedule on system development and performance is a reality that cannot be ignored. Program Management and Systems Engineering disciplines must consider the impact to the entire system environment before reacting to the pressures imposed by cost and schedule constraints.

There will be times when there appear to be few alternatives. Should a program scramble under extreme duress making critical decisions without substantive or proper data or justification. The political pressures are extreme as evidenced in the Osprey Case listed earlier in this paper. These decision tradeoffs could ultimately result

in a catastrophic event. Is it better to deliver a system on schedule with reduced safety, performance, and/or test margins? Tough decisions will have to be made and hopefully this paper will assist in understanding the risks involved.

Biography

I. Ronald Stroup, Software Safety and Certification Lead, Federal Aviation Agency, AIO 200, 800 Independence Avenue, S.W. Washington, DC. 20591, telephone – (202)-493-4390, facsimile (202)-267-5080, e-mail - ronald.l.stroup@faa.gov

Mr. Ronald Stroup joined the Federal Aviation Administration as an Aerospace Engineer in 1989. He holds a Bachelor of Science in Avionics Engineering from Parks College of Saint Louis University. Mr. Stroup served as a systems engineer in the Aircraft Certification Services' Chicago Aircraft Certification Office and in 1997 became the Software Technology Specialist for the Aircraft Certification Service. His responsibilities included providing technical expertise in the area of software approvals and acted as a focal point to improve the software approval process.

Since 1998, Mr. Stroup has served as the Software Safety and Certification Lead for the Office of Information Services and Chief Information Officer. His duties include developing and applying software assurance standards to the acquisitions of software-intensive National Airspace Systems.

Mr. Stroup is a member of the RTCA/SC-190 Committee, IEEE, and Co-Program Manager for the Streamlining Software Aspects of Certification initiative. He served as a Subject Matter Expert for the Software Fundamentals course to develop training in the application of RTCA/DO-178B Assurance Standard and software engineering practices. Mr. Stroup serves on the FAA's Systems Engineering Council, the System Safety Working Group and Eurocontrol's Software Task Force.

II. Warren P. Naylor, BAE SYSTEMS Applied Technologies, System Safety Manager, 1601

Research Boulevard, Rockville, MD 20910-3173, telephone – (240)-994-1765, facsimile - (202) -548-5504, E-mail - warren.naylor@baesystems.com

Warren P. Naylor is the Systems Safety manager and lead Systems Safety Engineer for BAE SYSTEMS Applied Technologies with over 8 years of software/systems safety experience and over 25 years of software development and field service support with Department of Defense weapon systems.

Mr. Naylor is currently providing software safety security, and human factors support to the Federal Aviation Administration - AIO 200. Additionally, Mr. Naylor is providing independent system safety consultation and safety management services to the NATO SEASPARROW Project Office (NSPO / PMS471) and manages the safety efforts for the Naval Surface Fire Support (PMS 529) programs.

Mr. Naylor's proximity to PMS471, the FAA , and NSWCDD G71 has provided him with a unique insight into implementing, managing, and maintaining safety in highly complex, integrated COTS-based real-time mission and safety critical systems.

Acknowledgements

We wish to thank all of our colleagues, both past and present, for their support, cooperation, advice, and most of all their friendship.

References

¹ Flaherty and Ricks., The Washington Post. Front Page `9 February 2001.

² Glass, R., Software Runaways. Prentice Hall PTR, 1998

³ Advance Automation System, (AV-1998-113, April 5, 19980 u.s. department of Transportation – Office of Inspector General Audit Report

⁴ Sawyer, The Washington Post, March 4, 2001.

⁵ Blanchard and Fabrycky., *Systems Engineering and Analysis*, 3rd Edition, 1998, Prentice-Hall Inc.

⁶ Story, N., *Safety-Critical Computer Systems*, Addison-Wesley, 1996

⁷ Perry, T., In Search of the Future of Ait Traffic Control. *IEEE Spectrum*, August 1997.

⁸ Leveson, N., *Safeware System Safety and Computers*. Edison-Wesley, 1995