

# Challenges in Software Aspects of Aerospace Systems

Kelly J. Hayhurst  
NASA Langley Research Center  
k.j.hayhurst@larc.nasa.gov

C. Michael Holloway  
NASA Langley Research Center  
c.m.holloway@larc.nasa.gov

## Abstract

*For many years, NASA Langley Research Center has cooperated with the Federal Aviation Administration (FAA) in research about software engineering methods for aerospace applications. Recent research has focused on software aspects of the FAA's certification process. In this paper, the results of the Streamlining Software Aspects of Certification (SSAC) program are examined to provide insight into current challenges in the aerospace industry in developing and assuring complex, software-based systems. We conclude that at the root of many of the current challenges lies the challenge of accurately communicating requirements between groups of people.*

## 1. Introduction

Aerospace systems now depend on software more than ever before to ensure safety and efficiency. In an Aviation Week & Space Technology commentary, David Hughes wrote "Information technology is becoming a key part of everything the aerospace and defense industry does for a living, and as the century closes it is computers and software that hold the keys to the future. The industry is being transformed from dependence on traditional manufacturing into something that looks more like IBM and Microsoft with wings." [1]

This transformation applies not only to industry, but also to government agencies such as the FAA and NASA. For example, software-based capabilities for communications, navigation, and surveillance for air traffic management (CNS/ATM), such as new controller aids like the Traffic Management Advisor, passive Final Approach Spacing Tool, and User Request Evaluation Tool, are envisioned to be the backbone of the FAA's new airspace system.

Within NASA, the reliance on software is growing quickly. According to former NASA Administrator Dan Goldin, 25 years ago the Voyager spacecraft had 5000 lines of computer code, whereas the International Space Station (ISS) has 1.4 million lines and the percentage of a space-mission budget devoted to software has risen from 5% to 20%, and will soon reach 50% [2]. NASA's

reliance on software is not limited to space flight. Many aeronautics research programs depend critically on software-provided capabilities for success. The Small Aircraft Transportation System (SATS) program is one example of such a program.

In short, NASA and FAA efforts to modernize the National Airspace System and space exploration cannot be achieved without massive amounts of software supporting safety-critical functions. Developing complex software-based systems and verifying that these systems meet safety requirements and assurance standards will be essential. Recent history shows, however, that accomplishing this task will be quite hard.

Examples abound of large, complex aerospace systems that have failed or overrun planned costs and schedules. Recent examples from civil aviation include the multi-million dollar budget overruns and multi-year delays in fielding the Wide Area Augmentation System (WAAS), the Standard Terminal Automation Replacement System (STARS), and the Airport Movement Area Safety System (AMASS) program. Table 1 shows changes in estimates for total program cost and scheduled operational deployment for these three programs as estimated by the Office of Inspector General in 2000 [3]. Note that the current estimated date for WAAS scheduled operation is 2003 [4].

**Table 1. Cost and schedule delays for FAA modernization programs**

| Project | Total Program Cost (in millions) |                  | Scheduled Date for Operation |                  |
|---------|----------------------------------|------------------|------------------------------|------------------|
|         | Original Estimate                | Current Estimate | Original Estimate            | Current Estimate |
| WAAS    | \$892.4                          | \$2,900.0        | 1998                         | 2000             |
| STARS   | \$940.2                          | \$1,400.0        | 1998                         | 2002             |
| AMASS   | \$59.8                           | \$151.8          | 1996                         | 2002             |

For all of these programs, software problems have been cited as contributors to the cost and schedule problems. For example, in testimony to Congress about WAAS, Gerald Dillingham of the United States General Accounting Office said: "Software development—the most critical component of key FAA modernization programs—has been the Achilles' heel of FAA's efforts

to deliver programs on time and within budget.” [5]

NASA is not immune to software difficulties. During a session of the International Space Symposium in October 2001, former Administrator Goldin was quoted as saying “Software is ripping apart this industry.”[6] Some of NASA’s most notable software woes are substantial budget overruns on the International Space Station [7], and failures in the Mars Climate Observer [8] and Mars Polar Lander [9] missions. In each of these cases, software problems contributed significantly to the failures and overruns.

In the rest of this paper, we describe work that NASA Langley has done to try to identify why software problems such as those cited are happening.

## 2. Streamlining Software Aspects of Certification Program

Motivated by the cost and schedule overruns attributed to software on major CNS/ATM projects, the FAA sponsored the Streamlining Software Aspects of Certification (SSAC) program. As part of this program, the FAA commissioned an independent team of software engineering and safety experts to determine whether the cost and time associated with the software approval process can be reduced without compromising safety. This team, which was designated the SSAC technical team, was lead by NASA Langley Research Center.

Although the FAA sponsors and the technical team members had some strong ideas about what the problems were, the team decided to try to find out directly from industry what their real problems were, instead of simply assuming that the team members’ ideas were correct. To obtain the desired information from industry, the technical team conducted two workshops and an extensive survey. The two workshops were held with aviation software industry representatives and certification authorities to identify major issues affecting cost, schedules, and software approval.

### 2.1. Workshops

During the first workshop (held in January 1998), participants identified more than 200 individual concerns about RTCA/DO-178B “Software Considerations in Airborne Systems and Equipment Certification” [10] (the *de facto* standard for development and assurance of software for commercial transport aircraft), and other aspects of the software approval process. All of these were documented and summarized into 14 general issues to be considered by the technical team for further data collection. The results of the first workshop were documented in a NASA Langley technical report [11].

A second workshop was held in May 1998 to determine the issues the industry participants considered to be the most important to study. Based on the results of this workshop, and recommendations from the FAA, the team decided to pursue further data collection in the following areas:

- Interpretation and application of software policy and guidance
- Cooperation between the FAA and industry
- Availability of information about the certification process
- The effectiveness of specific activities required by DO-178B, including independence, structural coverage, traceability, documentation, quality assurance, and tool qualification
- The relationship between DO-178B and safety
- The effectiveness of the designee (designated engineering representative, DER) system

The next step in data collection was to assure that the issues identified as most important to the workshop participants were indeed significant issues for the general population that develops airborne or ground-based systems containing software (DO-178B compliant software in particular).

### 2.2. Survey

Based on the workshop results, an extensive survey of the aviation software community was conducted. The survey was designed to determine overall satisfaction in the aviation industry with the FAA’s software approval process, and determine the extent and significance of the problem areas identified at the workshops.

A number of steps were taken to produce a good quality questionnaire, including consultation with the Center for Survey Research (CSR) at the University of Virginia. After the team drafted a questionnaire, the survey questions were reviewed by CSR to help remove bias in the way questions were stated and ensure that the response options were independent and complete. FAA representatives reviewed the questions to ensure that their concerns were addressed. Finally, two pretests were conducted with small subsets of the population. The pretest participants provided feedback on the questionnaire’s clarity and comprehensiveness.

The survey questionnaire contained over two hundred questions about the FAA’s software approval process and policy, technical aspects of software development (including questions about verification, quality assurance, and tool qualification), and safety. For the most part, the questions came directly from the issues raised at the two

workshops. The organization and content of the survey is shown in Table 2.

**Table 2. Organization and content of the SSAC survey questionnaire**

| Survey Section | Topic   |
|----------------|---|
| A & B          | Respondent Background and Experience                                  |
| C              | FAA Policy, Guidance, and Audits                                      |
| D              | Aircraft Certification Offices  |
| GD             | Approving Authorities for Ground-based Systems                        |
| E              | Independence  |
| F              | Modified Condition Decision Coverage                                  |
| G              | Traceability  |
| H              | Quality Assurance   |
| I              | Documentation   |
| J              | Tool Qualification  |
| K              | Safety  |
| L              | Designated Engineering Representatives                                |
| GL             | Using Designated Engineering Representatives for Ground-based Systems |
| M              | Availability of Information about the Certification Process           |
| P              | Appropriateness of DO-178B for Ground-based Systems                   |

Note that a survey of this size is not typically recommended, because completing a lengthy survey requires a 1-2 hour commitment from each respondent. However, the SSAC technical team believed that the survey population would be sufficiently motivated to complete the survey as a way to substantiate their concerns to the FAA.

The survey population included people with different levels of experience with RTCA/DO-178B, experience with a variety of airborne and ground-based aviation products, and experience with projects of various size and criticality. FAA representatives were not included in the survey population.

To help allay fears about commenting on an approving authority, CSR collected all of the responses and filtered identifiers, such as name and company affiliation. During the collection period, CSR received 300 questionnaires, for a response rate of approximately 72%. Of those 300 questionnaires, 292 were completed surveys suitable for analysis.

### 2.3. The Survey Results

The SSAC survey results point to a number of

technical challenges in software engineering, and challenges related to assuring software systems. Table 3 lists some of the findings from analysis of the survey.

**Table 3. Selected findings from the SSAC survey**

| Topic  | Finding   |
|--|---|
| Software policy and guidance                               | <p>Written software policy and guidance provided for the certification process appears to be inadequate in both availability and quality. Clarification of guidance in DO-178B is needed for the following subjects:</p> <ul style="list-style-type: none"> <li>• independence</li> <li>• modified condition/decision coverage (MC/DC),</li> <li>• source to object code traceability,</li> <li>• submittal of certification data,</li> <li>• requirements for tool qualification, and</li> <li>• derived requirements</li> </ul> <p>(Requests for clarifications on these subjects were forwarded to the RTCA Special Committee 190, which was established in 1996 to clarify unclear sections of DO-178B)</p> |
| Interaction between industry and certification authorities | <p>Communication problems exist between certification authorities and applicants, especially with respect to inconsistent interpretation of software policy and guidance, DO-178B in particular.</p>  |

Based on the survey results, the SSAC technical team identified the following seven concerns as being industry-wide concerns:

- Inadequate information is available about certification
- Inconsistencies exist within the FAA in interpreting and following policy and guidance
- Insufficient knowledge of software engineering and related disciplines exists within industry
- Insufficient knowledge of software engineering and related disciplines exists within the FAA

- Inadequacies, inconsistencies, and inefficiencies exist in the designee system
- Lack of cooperation exists between the FAA and industry
- Requirements definition is difficult

In its report on the survey, the team also made ten specific recommendations to the FAA based on the observations, and suggested areas for which additional data collection was needed [12]. The FAA's response to the recommendations is available on the internet [13].

### 3. Additional Analysis

Additional analysis was completed recently of the seven industry-wide concerns identified by the SSAC technical team. This analysis suggests that each of the concerns has the same basic root: challenges in communication. That is, each of the seven concerns is in some way a result of breakdowns in communication.

#### 3.1. Aviation Example

For example, consider the first two concerns identified by the SSAC technical team: inadequate information is available about certification; and inconsistencies exist within the FAA in interpreting and following policy and guidance. These concerns exist, at least in part, because ineffective communication exists about certification requirements. Ineffective communication alone cannot account for every instantiation of these two concerns, but it can account for many of them, and is a likely contributor to all of them.

As an illustration of ineffective communication about certification requirements, consider the following expression

(A and B) or (B and C) or (A and C)

where A, B, and C are simple Boolean variables.

To meet the DO-178B verification objectives for Level A — that is, the highest criticality — software, one must know the number of *conditions* in this statement. According to the glossary entry for *condition* in DO-178B, a condition is “a Boolean expression containing no Boolean operators.”

Given this information, how many conditions are in the expression?

Attendees at a recent FAA training course were asked this question, and told to choose among answers of 3, 4, 6 or 9 conditions. Fourteen attendees said 3, seven attendees said 4, sixteen attendees said 6, and two attendees said 9. The correct answer is 6 conditions.

When first presented with results such as these, many people's first reaction is probably to question the competency of those who did not know the answer. Such a reaction would be inappropriate, however. Determining the correct answer is by no means a simple task. It is not a simple task because of communication problems within DO-178B itself.

To know the correct answer, one must know that the full definition for *condition* is not contained in the glossary entry for that term. Instead, part of the definition is buried in the entry for *decision*, which reads as follows:

Decision: A Boolean expression composed of conditions and zero or more Boolean operators. A decision without a Boolean operator is a condition. If a condition appears more than once in a decision, each occurrence is a distinct condition.

The last sentence in this glossary entry is an essential part of the meaning of the term *condition*. Applying this sentence to the expression shown previously, and reading the expression for left to right, we can identify the conditions as follows: the first A, the first B, the second B, the first C, the second A, and the second C.

If an industry software engineer asked several FAA engineers this question, and received several different answers (which appears to be likely to happen), the software engineer is likely to attribute this to “inconsistencies within the FAA in interpreting and following policy and guidance.” Inconsistency certainly exists. Its existence is guaranteed by the way the policy and guidance (DO-178B in this case) is written.

In our particular example, inconsistency — that is, different interpretations by different people — is guaranteed for three reasons. First, inconsistency is guaranteed by the distribution of the definition for *condition* across two glossary entries in a way that is not natural. Unless there are words one does not understand within the definition of a particular term, one does not expect to have to look outside the term's glossary entry to learn its meaning.

Inconsistency is further guaranteed by the use of terms with strong connotations in ways that violate those connotations in many people's minds. Almost any software engineer who reads DO-178B will already have formed mental models of how terms such as *condition* and *decision* are used within software engineering. For many people, these mental models will not correspond to the way these terms are used in DO-178B. This makes understanding difficult [14].

The third guarantee of inconsistency is the lack of clarifying guidance or educational material until recently. Although DO-178B was published in 1992, useful

clarifying material was not published until the last two years. This material includes two reports from RTCA/SC-190 [15, 16] and a NASA Langley technical report [17].

With these three guarantees of inconsistency existing — all three of which are communication problems — no one should be surprised that different FAA engineers give different answers to questions such as the one posed in the example. Any group of highly skilled and intelligent people will inevitably do the same in similar situations.

Examples such as this one could be given for the other five SSAC-identified concerns, too. Examples alone do not constitute proof, but they do lend support to our assertion: challenges in communication appear to be at the root of many of the concerns that the industry and the FAA have with software aspects of commercial aviation systems.

### 3.2. NASA Example

An analogous assertion seems applicable to NASA, too: challenges in communication appear to be at the root of many of the concerns that the industry and NASA have with software aspects of space systems. Although nothing similar to the SSAC workshops and survey has been done within NASA, some evidence exists to support the assertion.

Consider, for example, the Mars Climate Orbiter failure. The Mishap Investigation Board “determined that the root cause for the loss of the MCO spacecraft was the failure to use metric units in the coding of a ground software file ... used in trajectory models. Specifically, thruster performance data in English units instead of metric units was used in the software application code.” The Board determined that the existing software interface requirements document specified the use of metric units, and noted: “the trajectory modelers assumed the data was provided in metric units per the requirements.” [18]

The Board’s report makes clear that communication problems played a significant role in the failure. “Inadequate communications between project elements” was explicitly designated as a “contributing cause.” Also, the Board stated in the Executive Summary of the final report: “Most mission failures and serious errors can be traced to a breakdown in existing communication channels, or failure to follow existing processes....” [18]

Researchers and practitioners of mishap investigation and reporting may disagree with the way the Board used terms such as “root cause” and “contributing cause”, although the Board’s usage is consistent with the applicable NASA guideline [19]. Such disagreements are not important for the purposes of this paper. What is

important is simply that difficulties in communication existed, and contributed to a mission failure.

## 4. The Communication Challenge Simplified

Difficulties in communication can occur in many different ways, and in many different situations. The examples cited above showed communication difficulties in writing documents, in interpreting documents, in resolving ambiguities, and in interactions between people in different parts of a project. Difficulties in other types of communication can be easily imagined.

Attempting to address directly and separately each and every different instantiation of communication problems would be hard, and unlikely to succeed. Fortunately, the myriad of communication challenges can be simplified, as is shown in Figure 1.

The majority of communication necessary for developing aerospace software systems occurs along two main channels and involves communicating two main types of information.



Figure 1. Communication channels simplified

One communication channel exists between regulatory people and systems people. *Regulatory people* refers to the people whose primary responsibility is to certify that a particular system may be used. For commercial aviation in the United States, this would be the FAA. For NASA space missions, it would be the people who must agree to allow a mission to proceed. *Systems people* refers to the people whose primary responsibility is to build a particular system.

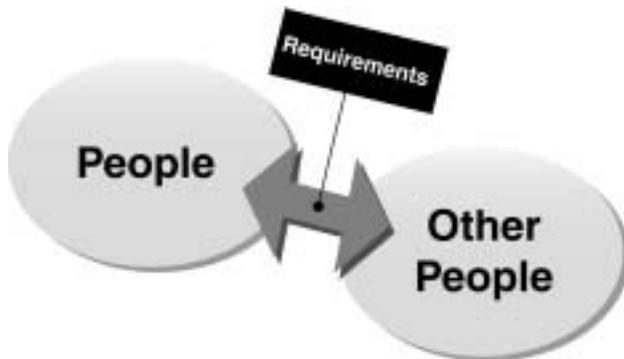
The primary content of the communication between these groups is certification requirements. The regulators and system developers must both understand the requirements the system must satisfy for it to be approved, and how the system will be shown to satisfy these requirements. Reaching the needed level of mutual understanding involves extensive communication between the two groups.

A second major communication channel exists between the systems people and the software people. The

*software people* are those people whose primary responsibility is to build the software aspects of a particular system.

The primary content of communication between these groups is technical requirements. Technical requirements are not the only thing about which these groups must communicate, but they are the primary thing. For example, the systems people and software people must agree on what system functions will be implemented in software, and on the specifics of what those functions must accomplish.

Note that considerable overlap exists in the salient characteristics of these two communication channels and content. Both channels involve groups of people with different primary responsibilities. The content flowing through both channels consists primarily of requirements that must be satisfied. This observation of overlap suggests that communication challenges may be further simplified, as shown in Figure 2.



**Figure 2. Further simplification**

Herein lies the foundational challenge in software aspects of aerospace systems: communicating requirements between groups of people consistently, completely, concisely, and promptly.

## 5. Implications

If, as we believe, the foundational challenge in software aspects of aerospace systems involves communicating requirements, then several important implications follow. Four of these implications are discussed below.

The first implication is this: improving the communication of requirements is essential for real progress in efficient development of safe and reliable aerospace systems. Making real progress without attacking a root problem simply is not possible. At best, attacking other problems will yield only marginal progress.

Another implication from the foundational status of communication challenges is that research efforts should concentrate here. The growing number of researchers in the field called "requirements engineering" suggests that many people are beginning to understand the importance of requirements in systems development.

To date, existing requirements engineering work has tended to focus heavily on technical requirements. Quite a bit less effort has focused on certification requirements. A third implication from the analysis presented here is that work is needed on both types of requirements. Further, if the simplification shown in Figure 2 is valid, then most of the existing work should be as applicable to certification requirements as it is to technical requirements. All that should be needed are researchers and practitioners willing to make the applications.

The fourth, and perhaps most important, implication from the primacy of communications is that engineers cannot expect to make significant progress without active cooperation with people from other disciplines. One particular discipline that seems likely to be able to make important contributions is linguistics [20].

## 6. Summary

This paper began by noting that aerospace systems now depend on software more than ever before to ensure safety and efficiency. Reliance on software is continuing to grow, with no end in sight. At the same time, evidence is continuing to grow that efficiently developing safe software systems is fraught with significant challenges, with no solutions in sight.

Based on data collected during the FAA-sponsored Streamlining Software Aspects of Certification project, we believe that at the root of many of these significant challenges lies the challenge of accurately communicating requirements between groups of people. Effectively meeting this challenge is vital to the success of future aerospace vehicles, systems, and missions.

## 7. References

- [1] D. Hughes, "Information Technology: This Changes Everything," *Aviation Week & Space Technology*, December 21/28, 1998.
- [2] P. Guinnessy, "Goldin Maps NASA's Past, Present, and Future," *Physics Today*, vol. 54, no. 4, April 2001, pp 25-26.
- [3] Office of the Inspector General, "Modernizing the Federal Aviation Administration: Challenges and

Solutions," Washington, D.C. #AV-2000-039, February 17, 2000.

[4] Federal Aviation Administration, "Implementation Status", last update date unknown, at [http://gps.faa.gov/Programs/WAAS/Implementation\\_Status/implementation\\_status.htm](http://gps.faa.gov/Programs/WAAS/Implementation_Status/implementation_status.htm), accessed on December 11, 2001.

[5] United States General Accounting Office, "National Airspace System: Problems Plaguering the Wide Area Augmentation System and FAA's Actions to Address Them," Statement of Gerald L. Dillingham, Associate Director, Transportation Issues, Resources, Community, and Economic Development Division before the Subcommittee on Aviation, Committee on Transportation and Infrastructure, House of Representatives GAO/T-RCED-00-229, June 29 2000.

[6] J. Faust, "Goldin challenges NASA, aerospace industry", spaceflightnow.com, October 31, 2001, at <http://spaceflightnow.com/news/n0110/31goldin/>, accessed on October 31, 2001.

[7] F. Moring, Jr., "ISS Cost Growth May Continue," *Aviation Week & Space Technology*, April 9, 2001, p. 39.

[8] M. A. Dornheim, "Faulty Thruster Table Led to Mars Mishap," *Aviation Week & Space Technology*, October 4, 1999, p. 40.

[9] JPL Special Review Board, "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions," Jet Propulsion Laboratory, JPL D-18709, March 22, 2000.

[10] RTCA Inc., "Software Considerations in Airborne Systems and Equipment Certification," Washington, D.C. RTCA/DO-178B, 1992.

[11] K. J. Hayhurst, C. M. Holloway, C. A. Dorsey, J. C. Knight, N. G. Leveson, G. F. McCormick, and J. C. Yang, "Streamlining Software Aspects of Certification: Technical Team Report on the First Industry Workshop," NASA Langley Research Center, Hampton, Virginia, NASA Technical Memorandum TM-1998-207648, April 1998.

[12] K. J. Hayhurst, C. A. Dorsey, J. C. Knight, N. G. Leveson, and G. F. McCormick, "Streamlining Software Aspects of Certification: Report on the SSAC Survey,"

NASA Langley Research Center, Hampton, Virginia, NASA Technical Memorandum TM-1999-209519, August 1999.

[13] Federal Aviation Administration, "FAA's Response and Current Supporting Activities", September 13, 1999, at <http://shemesh.larc.nasa.gov/ssac/faa-response-letter.pdf>, and <http://shemesh.larc.nasa.gov/ssac/faa-response-to-survey.pdf>, accessed on December 6, 2001.

[14] C. Potts, "Metaphors of Intent," presented at RE'01: Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, 2001.

[15] RTCA Inc., "Second Annual Report for Clarification of DO-178B 'Software Considerations in Airborne Systems and Equipment Certification'," Washington, D.C. RTCA/DO-248A, September 13, 2000.

[16] RTCA Inc., "Final Report for Clarification of DO-178B 'Software Considerations in Airborne Systems and Equipment Certification'," Washington, D.C. RTCA/DO-248B, October 12, 2001.

[17] K. J. Hayhurst, D. S. Veerhusen, J. J. Chilenski, and L. K. Rierson, "A Practical Tutorial on Modified Condition/Decision Coverage," NASA Langley Research Center, Hampton, Virginia NASA/TM-2001-210876, May 2001.

[18] NASA, "Report on Project Management in NASA: Phase II of the Mars Climate Orbiter Mishap Report", Mars Climate Orbiter, Mishap Investigation Board, NASA Headquarters, March 13, 2000, at [ftp://ftp.hq.nasa.gov/pub/pao/reports/2000/MCO\\_MIB\\_Report.pdf](ftp://ftp.hq.nasa.gov/pub/pao/reports/2000/MCO_MIB_Report.pdf), accessed on December 6, 2001.

[19] NASA, "NASA Procedures and Guidelines for Mishap Reporting, Investigating, and Recordkeeping," Safety and Risk Management Division, NASA Headquarters, Washington, D.C., NPG: 8621.1, June 5, 2000.

[20] K. S. Hanks, J. C. Knight, and E. A. Strunk, "Erroneous Requirements: A Linguistic Basis for Their Occurrence and an Approach to Their Reduction,," presented at 26th Annual NASA Goddard Software Engineering Workshop, Greenbelt, Maryland, 2001.