The Application of Multiple Team Inspections on a Safety-Critical Software Standard

Leonard L. Tripp, William F. Struck, and Bryan K. Pflug
The Boeing Company

Abstract:
This paper discusses the application of a multiple team inspection in the revision of a software standard which is applied to the procurement of safety-critical systems used on commercial airplanes.

I. INTRODUCTION
The development of a standard typically involves the following steps:
1. Need for a standard identified.
2. Development of draft standard authorized.
3. Draft standard prepared.
4. Draft reviewed by qualified, technical persons.
5. Standard approved for publication.

This paper discusses the application of multiple team inspections to improve the technical review and quality of a safety-critical software standard. Inspections provide a structured and manageable discipline to an often ad-hoc technical review process.

The discussion includes material on the background of the standard and its revision, the inspection process, the application of a multiple team approach to inspections and results, lessons learned, and conclusions.

II. BACKGROUND ON THE STANDARD
The safety-critical software standard is applied to the procurement of digital avionics computer systems developed by avionics suppliers for installation by our company in commercial airplanes. The resultant systems and associated documentation must satisfy regulatory agency (e.g., FAA) certification requirements [1].

The standard is a comprehensive document that addresses all "phases" of software development from analysis of system requirements through verification and certification support. It represents the experience and opinions of a group of experts for the "best" or preferred practice for the development of avionics software.

The first version of the standard was released in 1983. In the intervening years, it had been updated 3 times with resulting changes that were minor relative to this revision. Reviews of earlier versions shared the characteristic of having multiple iterations with no established criteria for acceptability and section completion, minimal "teaming" of the results, and little progress on resolution of the open issues. The current revision team had developed a mission statement defining the revision's requirements and completion metrics. The mission stated:

- Develop company requirements for the software development processes and results of those processes to be applied in the procurement of avionics and related software for commercial airplanes.

These requirements were to:
- Be derived from the existing baseline documents and combined into an integrated set of requirements in one volume.
- Include supplier requirements for compliance to the standard and be a mechanism to facilitate monitoring of compliance.
- Establish content, not format, requirements for results.
- Establish and facilitate incremental development and delivery of results, and address special issues.
- Focus on the avionics development process by:
  - identifying the sequence and content of events and ensuring each requirement is understandable, reasonable, contractually enforceable, unique, uniquely identifiable, verifiable, and accompanied by rationale when appropriate.

The revision process included preparation of a draft of the standard's sections which was performed by members of the revision team. The draft sections were subjected to preliminary reviews. After the content of the sections was stable, an integration activity was required to make the standard consistent. Then the draft standard was distributed for a broader technical review. Comments received were resolved by a Change Control Board (CCB) before approval and release of the revision for use on the airplane programs. The CCB, consisting of airplane program software management, was responsible for making the final decisions on the standard's content.

III. INSPECTIONS
The inspection approach is based on early work at IBM [2] and further development by Michael Fagan [3]. During the
limited technical review of the standard's sections, the inspection approach was applied to a portion of the software standard. Encouraged by the results, it was determined that the entire standard would benefit from an inspection. The use of multiple teams was motivated by work done at the University of Minnesota [5].

For descriptions of the roles and activities of the inspection process, see [4] and [6]. Particulars on a multiple team inspection application can be found in [5] and additional background on others' experiences with inspections and defect prevention in [7], [8], and [9].

IV. THE MULTIPLE TEAM APPLICATION

The Teams

To facilitate ownership of the revision and obtain valid results, team members from all airplane programs were included in the inspection effort. This necessitated conducting the inspections in 4 locations separated by up to 30 miles. Software managers were invited to select personnel from their programs to participate. To round out the perspective of the teams, participants from disciplines other than software were also invited.

The 5 teams consisted of 4 to 8 inspectors per team, with variable levels of experience. Table 1 illustrates the experience levels of the teams. There was also a wide range of disciplines represented, including system engineers, software engineers, buyers, system acquisition managers, chief engineers, quality improvement specialists, quality assurance personnel, and suppliers.

<table>
<thead>
<tr>
<th>Table 1 Team Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team</td>
</tr>
<tr>
<td>Number of Members</td>
</tr>
<tr>
<td>Average - Years with Company</td>
</tr>
<tr>
<td>Range - Years with Company</td>
</tr>
<tr>
<td>Average - Years of Software Experience</td>
</tr>
<tr>
<td>Range - Years of Software Experience</td>
</tr>
</tbody>
</table>

Preparation and Distribution of Materials

Preparation for the inspections included training and kickoff meetings. Training consisted of a short course which included a high level view of the inspection process, inspection roles, and rationale for the rules of inspections.

The kickoff meetings were held to distribute the current draft of the standard, the inspection standard and checklist, and other applicable standards and policies. Figure 1 is a sample inspection standard and figure 2 is an example of an inspection checklist. Also, roles were assigned (standards expert, backward reader, etc.) and questions the inspectors had regarding the process were addressed.

The draft was "chunked" into 3 pieces (each from 12-16 pages in length) to reduce the amount of material to be inspected for each logging session and to ensure equivalent coverage of all sections.

<table>
<thead>
<tr>
<th>Figure 1 Sample Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The standard shall be consistent and compatible with company policy, agreements, and practices.</td>
</tr>
<tr>
<td>2. The document shall specify all review points and their entry and exit criteria in the software life cycle.</td>
</tr>
<tr>
<td>3. Each requirement shall be stated only once.</td>
</tr>
<tr>
<td>4. The standard shall provide content and schedule requirements for all deliverables.</td>
</tr>
<tr>
<td>5. &quot;How-to&quot; requirements shall not be permitted.</td>
</tr>
<tr>
<td>6. Each requirement in the standard shall be stated such that the evidence of compliance can be determined by a clerk.</td>
</tr>
<tr>
<td>7. Each requirement in the standard shall use the verb, &quot;shall.&quot;</td>
</tr>
<tr>
<td>8. All unique terms shall appear or be referenced in the glossary.</td>
</tr>
<tr>
<td>9. Terms shall be used consistently throughout the standard.</td>
</tr>
<tr>
<td>10. The standard shall use proper grammar.</td>
</tr>
<tr>
<td>11. The standard shall use proper punctuation.</td>
</tr>
<tr>
<td>12. All acronyms shall be listed in the acronym list.</td>
</tr>
<tr>
<td>13. All terms and phrases of interest shall appear in the index.</td>
</tr>
</tbody>
</table>

Inspector Preparation for the Meetings

It was estimated that each inspector would need to commit about 2 hours to prepare before each defect logging meeting. Preparation activities include reading related materials, inspecting the standard while recording potential defects, classifying each defect discovered, and counting the number of defects detected.

Most inspectors were able to prepare within the estimated time. Several inspectors required 3-4 hours of preparation and two required 5-7 hours for each session.

The Logging Meetings

The two hour logging meetings consisted of handling administrative issues (e.g., recording preparation time and
number of defects), defect logging, and defect causal analysis. The moderator controlled the pace and focus of the meetings: he determined when breaks and causal analysis were required and determined the segment (page, paragraph) open for the logging of defects (i.e., all defects for a particular segment were recorded before recording defects of the next segment).

<table>
<thead>
<tr>
<th>Figure 2 Sample Checklist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is the requirement <strong>necessary</strong>?</td>
</tr>
<tr>
<td>2. Is each requirement <strong>compatible</strong> with other requirements in the standard?</td>
</tr>
<tr>
<td>3. Can each requirement be implemented with available technology?</td>
</tr>
<tr>
<td>4. In your opinion, will the requirement provide the desired result?</td>
</tr>
<tr>
<td>5. Is there only one possible interpretation of each requirement?</td>
</tr>
<tr>
<td>6. Is each standard statement clear, complete, correct, and concise?</td>
</tr>
<tr>
<td>7. Are there any requirements missing?</td>
</tr>
<tr>
<td>8. Is sufficient rationale provided for the requirement?</td>
</tr>
<tr>
<td>9. Do the requirements <strong>promote</strong> continuous quality improvement?</td>
</tr>
<tr>
<td>10. Is each requirement of the standard <strong>contractually enforceable</strong>?</td>
</tr>
<tr>
<td>11. Do the requirements <strong>promote</strong> safety? reliability? maintainability?</td>
</tr>
<tr>
<td>12. Is the standard <strong>style</strong> appropriate and consistent?</td>
</tr>
<tr>
<td>13. Do the standard section headers, titles, and figure titles accurately reflect the contents/text they describe?</td>
</tr>
<tr>
<td>14. Does the usage of a unique term match its <strong>definition</strong> in the glossary?</td>
</tr>
</tbody>
</table>

Defects were logged in a round-robin fashion until all defects for a segment were recorded. Logging consisted of a vocalization by the inspector of the location of each defect, its type and severity, and a brief description of the defect. The scribe was responsible for recording and clarifying the defects logged and collecting the mark-ups or notes from the inspectors at the end of the meeting.

The moderator placed emphasis on maintaining a high rate of defects logged per logging minute during the meetings. Rates varied from 3-6 defects/minute. This helped to ensure that all defects for the segment being inspected were logged and aided in preventing the inspectors from getting into long discussions of the issues and "fix-it" sessions. The primary intent of the inspections was to detect potential defects, not to correct them.

**Results of the First Application**

A sampling of defects was analyzed for 3 sections of the standard. The sampling represents approximately 20% of the standard's content. The sections analyzed were an introductory section, a section of programmatic requirements, and a section of technical requirements. Figure 3 shows the number of defects discovered by each team and the average number of defects for all five teams for the sampling.

The difference between the number of defects for team 1 and team 2 can be explained by two factors: (1) two of the inspectors of team 1 spent in excess of 6 hours preparing for each session and (2) the perspectives of the 2 teams was different (team 2 focused on detecting "substantive" defects while team 1 attempted to identify all defects, no matter how trivial). The number of defects for team 1 in the figure would be closer to the average if only "substantive" defects had been counted.

How many teams should be used for an inspection of a software standard? To answer this question, we calculated the average number of defects detected by 1 team, 2 teams, 3 teams, etc. Figure 4 illustrates the average number of **unique** defects detected by the teams for the sampling. The results for the 5 teams shows only a minor reduction in the number of unique defects detected. These results differ
with the results of others [5]. The "N-Fold Inspection" paper suggested less additional unique defects found with the addition of more teams. Our results show a near linear growth of unique defects detected by the addition of more teams.

The "N-Fold Inspection" paper suggested less additional unique defects found with the addition of more teams. Our results show a near linear growth of unique defects detected by the addition of more teams.

The sections sampled represent both programmatic (volatile) and technical (non-volatile) requirements, yet there is negligible difference in the rate of defect redundancy between sections. Figure 5 show the overall defect redundancy as a percentage of the total defects logged for the sampling.

This may be attributed to several factors:

- Lack of a reliable "oracle," which defines what should be contained in the standard. (The assumption being that the "N-Fold Inspection" teams [5] were inspecting an item for which there were established standards whereas a standard must provide for a range of interpretations and a balanced phrasing of requirements.)

- Combination of technical and programmatic requirements (which are determined by business and policy issues) rather than just technical issues.

- Inspectors were assigned roles to assume while inspecting (although they were not limited to detecting only those defects associated with their role).

- Inspectors with industry experience and expertise rather than "student" inspectors.

This low redundancy indicates an interesting phenomenon. All teams were provided with the same training and preparation materials (including the inspection standard and checklist) and moderated in a similar fashion. The teams were independent and isolated during the inspections (i.e., the results of one team were not shared with the other teams). The sections sampled represent both programmatic (volatile) and technical (non-volatile) requirements, yet there is negligible difference in the rate of defect redundancy between sections. Figure 5 show the overall defect redundancy as a percentage of the total defects logged for the sampling.

The overall redundancy rate of 14% is broken down in Figure 6. The "2" indicates the percentage of redundant defects detected by exactly two teams, the "3" indicates the percentage of redundant defects logged by exactly 3 teams, and so on for "4" and "5."

Causal Analysis

Causal analyses revealed several improvements which were incorporated during the process and into the resulting standard:

- The right personnel were identified and brought together to discuss and resolve the open issues.
Discussion of the open issues was bounded to the topic.
Training was provided to company engineers to promote consistent understanding.
New concepts were incorporated into the standard and greater flexibility allowed in applying the standard.
Scope of the standard was clarified to define its applicability and implementation.
Considerations for alternative business relationships were added to the standard.
Suggested wording was forwarded to and coordinated with the organization responsible for system acquisition standards.

The multiple team inspections prove to be beneficial for gathering input and resolution of the "major" issues from a variety of disciplines in a structured and manageable way.

V. A SECOND APPLICATION

Multiple team inspections were also applied to an Ada design standard. This standard is invoked for avionics software in conjunction with the standard of the first application as additional requirements related to the use of the Ada programming language. The standard contains design and coding requirements intended to promote the development of quality Ada software and to limit suppliers to Ada features appropriate for use in avionics software.
The application consisted of two teams with 6 inspectors on each team. The inspectors included some of those involved in the first application (three on 1 team and two on the other). Both teams' membership averaged 7.5 years of company experience and 4-5 years of avionics software experience.
The results indicate a similar redundancy rate of defects (16%) to the first application's redundancy rate (14% for sampled sections). This second application is mentioned because its results are consistent with and validate the redundancy results of the first application.

VI. LESSONS LEARNED

To ensure successful inspections, the following must be established and agreed on prior to beginning the inspection process:
- The importance of inspection item and level of inspection (single vs multiple team, number of teams).
- Identification of the policy and business issues that may affect the result.
- Involvement of all affected parties to promote ownership in the result.
- The purpose and scope of the inspection item.
- The purpose and scope of the inspection (standard and checklist).
- Extent of training needed (for novice inspectors).
- The ground rules for the inspection.
- Method of moderation (defects per minute vs. promoting team synergy).
- Inspectors' roles and responsibilities.
- Author/re-worker's role and responsibilities.

VII. CONCLUSIONS

Inspections and multiple team inspections (where the importance of minimizing residual defects is great) provide an efficient and manageable means of detecting defects and gaining resolution of those defects. The benefits of both are many and varied.
The benefits of inspections include:
- Improvement in the quality and integrity of development results.
- Reduced iterations of reviews.
- Improvement in customer satisfaction.
- Effective means of involving those who must approve the result.
- Effective framework for gathering feedback.
- Fast learning curve for novices to company policy and other disciplines.
- Inspectors become better authors.
- Inspections resulted in better relations with authors and users of standard.

In addition to those listed above, the benefits of multiple team inspections include:
- Range of inputs from various disciplines allows a higher level of confidence in the validity of the standard.
- Establishes credibility of the revision process with the users and acceptance of the result.
- Number of unique inputs indicates that, although the standard may not be perfect, a high percentage of the major defects were resolved.
- The standard's flexibility and adaptation of actual practice allow a wide range of applications and business implementations.
- Visible improvements in the result and its application.

There are number of benefits in conducting the logging meetings rather than having the inspectors just send their defect lists/marked-up copies to the re-worker (as may occur with a typical review scenario):
- Meetings are excellent way to transfer knowledge and experience between disciplines and experience levels, especially for educating newer employees in company policy and practices.
Inspections are especially appropriate at the early stages of a development, (for example, a system or software requirements document) when it is important to remove as many defects as possible so that the defects are not perpetuated to the design and implementation of the end item. They have been made a standard part of the process in our organization.

Multiple team inspections provide the means of detecting, correcting, and preventing a high percentage of defects from critical results. The resources required are costly, but provide substantial and justifiable benefits. The authors were surprised at the low rate of redundancy in defects between the teams, as we had estimated that there would be 50-60% redundancy.

The real test of how successful the inspection effort was for the software standard is how well the resulting standard works in the actual application. This includes how well it works for company engineers who monitor the supplier's development, and how well it works for the supplier in terms of applying the software standard to developments. Monitoring the application of the standard to future software procurements will be an ongoing activity which may further indicate the effectiveness of inspections.

VIII. OPEN ISSUES

There are three major issues that we feel require more study. How does one determine the number of teams most appropriate for various types of items to be inspected? How can the entry criteria be applied to ensure an item is of adequate quality to be inspected? How can exit criteria be applied to determine when sufficient quality is attained or if re-inspection is necessary?

Determining the right number of teams for inspecting an item is dependent on several things, including the importance of the item, the resources available, and motivation for the inspection. Our results indicate this can be a difficult decision.

Entry criteria must be established and applied which define minimum criteria that an item must meet before it is "ready" to be inspected. This ensures maximum return on investment being realized from the process.

The exit criteria for an item and the procedures for attaining them must also be considered. Following the inspection, the software standard was distributed to a more widespread review process that resulted in a number of change requests. All change requests were reviewed and resolved by the CCB before approval and release of the standard. This exercise was demanding but necessary to resolve the issues identified in the inspections and further confirmed by the change requests.

It is the opinion of the authors that the standard was "ready" for inspection following resolution of the change requests by the CCB. Many of the issues were business and policy issues that needed to be resolved because of their effect on the standard and its enforcement. The inspections raised the conscientiousness of those involved so that the issues were researched and appropriate philosophy and policy adopted. Initial feedback on the improvement of the quality of the standard revision has been positive.

The regulatory agency guideline for avionics software [1] is currently being revised and due to be released in 1-2 years. Because of the impact of the guideline on the software standard, the standard will again be revised following release of the guideline. It is anticipated that inspections will play a major role in the future revision of the software standard.

IX. ACKNOWLEDGEMENT

The authors wish to thank their management for the resources and support for the inspections, especially Roger Seeman, Chief Engineer of Avionics/Flight Systems Software Engineering. We also extend our thanks to the inspectors who participated, and their management for making them available.

X. REFERENCES