Building a CM Database: Nine Years at Boeing

Susan Grosjean
Boeing Electronic Products

A Boeing organization developed an Oracle-based database to track problems during the life cycle of the Boeing 777 airplane. Over nine years, it has evolved from mainframe to web implementation as technology has become available. This article reviews some basic approaches to developing configuration management databases and includes resulting lessons learned.

The Need for a Database System

Boeing Electronic Products (EP) designs and develops some of the avionics[1] for Boeing Commercial Airplanes (BCAG).

About 1990, when the 777 airplane program was getting started[2] , EP was instructed to develop an electronic database to track and record problems encountered during design and development of EP electronics. BCAG wanted a database it could access. After the 777 was fielded, all airplanes were to use this database.

The existing problem reporting system was paper-based. One paper copy of each problem report (PR) was stamped "original." Multiple copies were made for each person assigned to work the problem. Each person would record his or her response by writing on the hard copy. Periodically, the Integrated Product Team (IPT) leader, responsible for a given piece of avionics, would call together all members of the team, known as the Engineering Review Board. These meetings could last for hours, trying to take all inputs and create one original PR. An easier way was needed to consolidate board members' input.

Requirements Definition and Implementation

Engineers, IPT leaders, and other potential customers did not seem to know what they wanted/needed in a problem reporting system. All the organizational requirements were surveyed (EP Configuration Management Plan, RTCA/DO-178B "Software Considerations in Airborne Systems and Equipment Certification," EP Software Quality Assurance procedures, etc.). Using the PR as a guide, the existing process was captured in a flowchart representation with a text description of each block. A requirements document was developed from this process document.

The numbering scheme for the PRs was used in conjunction with automation of the problem reporting process. A two-field scheme was used. The first field was an Avionics Product (LRU)[3] designation such as 7WEU (777 Airplane Warning Electronics Unit). The IPT leaders and auditors needed a list of all PRs for a desired component. A search on this first field would provide that list. The second field was a four-digit number that was assigned sequentially. Once a PR receives this two-field dash number, it cannot be deleted, so there is no break in the numbering sequence. This simplified the IPT leader's control and made any audit process easier.

Four text fields, as well as signature blocks, were associated with each PR. Each text field needed to accommodate extensive data (20,000 characters each). In the PR heading were smaller fields for pertinent data needed for tracking and reporting. The entry of valid data to these fields was enforced through the use of pull-down lists that provided all valid options for a given field.

Because of the size and quantity of PRs, the speed and capacity of a mainframe was needed. EP chose to use Oracle on a VAX mainframe rather than an IBM due to the availability of VAX programmers in EP. EP was unable to find an existing user interface to meet its requirements. User access to the mainframe was via dumb monochrome monitors or by loading an access icon to the personal computers.

A major implementation complication was the need to have five variations of the PR (i.e. variations on the fields and associated process). The struggle to agree on a uniform PR form and process was futile; each user organization wanted its own little corner of the world. The result was five different PR databases. Maintaining requirements, processes, procedures, updates, and training for five different databases was a nightmare.

Throughout the PR process (see Figure 1), e-mail messages were to be triggered to inform personnel and to assign tasks. There are now 43 possible messages for each PR. For instance, after an originator of a PR has completed the appropriate "title" and "description" fields, the system sends the PR as an e-mail message to those responsible for the product (component) in question. The length of these e-mail messages was a challenge, since they included the 20,000-character-length text fields. This will be solved by e-mailing only a brief message that includes a URL access to the web site that the assigned can access to see the complete PR and perform the applicable function.

The system also automatically changes the status of the PR as it moves through the phases shown in Figure 1. Portions of a PR are approved in each phase. After approval, the status changes and fields are locked, preventing changes by unauthorized persons. Engineers, configuration management specialists, and IPT leaders have different levels of authorization.

User Acceptance of the System

After implementation of the requirements and extensive testing, the database was given to the users. A program directive and a users manual were written. Hands-on training was also conducted—not an intensive course, but intended to acquaint the user with the database and convey some of the rationale used for its creation.

Function Keys

The users' first reaction was, "This database is not user friendly." No one liked using the function keys to move around the database. Many keyboards did not support the function keys we were using, and corresponding control keys, `^B` or `F10`

= Exit had to be developed. These key strokes were displayed at the bottom of each screen.

Unlike the paper PR, the electronic version forced the entry of required information. The IPT leaders did not like the fact that the database forced everyone to do business the same way, and that it had checks and balances in place to ensure this practice.

But users liked the information the system provided: the position of a given PR in the life cycle, e-mail notification of PR status, who was assigned to work PR related tasks, and visibility into process bottlenecks. If a PR had languished on someone's desk for months, the IPT leader needed to know that. Sometimes the IPT leader was surprised to find that the languishing occurred on his or her own desk. Canned reports included listings of all PRs by responsible IPT leader or by assignee, and totals of the quantity of PRs by open and closed status.

Windows

Three years ago, Oracle offered Developer/2000, allowing a more user-friendly front end. We needed to retain the VAX, as by now all four text fields had increased to 65,000 characters each. With high-level management backing during the implementation of the front end, all Seattle-based groups adopted a single PR form. The groups also agreed with using basically the same process.

The Parts Engineering Organization in Seattle levied additional requirements, as it wanted a way to track and record problem reports involving obsolete parts.

Since a single part obsolescence is often common to several LRUs, a process enhancement was created to generate multiple corresponding PRs. A parent PR (the first PR written) is created and duplicate copies are made for each impacted LRU. These child copies retain a link to the parent. When the child PRs are closed, the parent PR can be closed.

Requirements expanded beyond Seattle to Irving, Texas—our manufacturing site. Irving has a problem-reporting system also implemented in Oracle. A requirement to transmit engineering problems between the Seattle and Irving databases was accomplished by developing an interface that transfers the PR from their database to Seattle's.

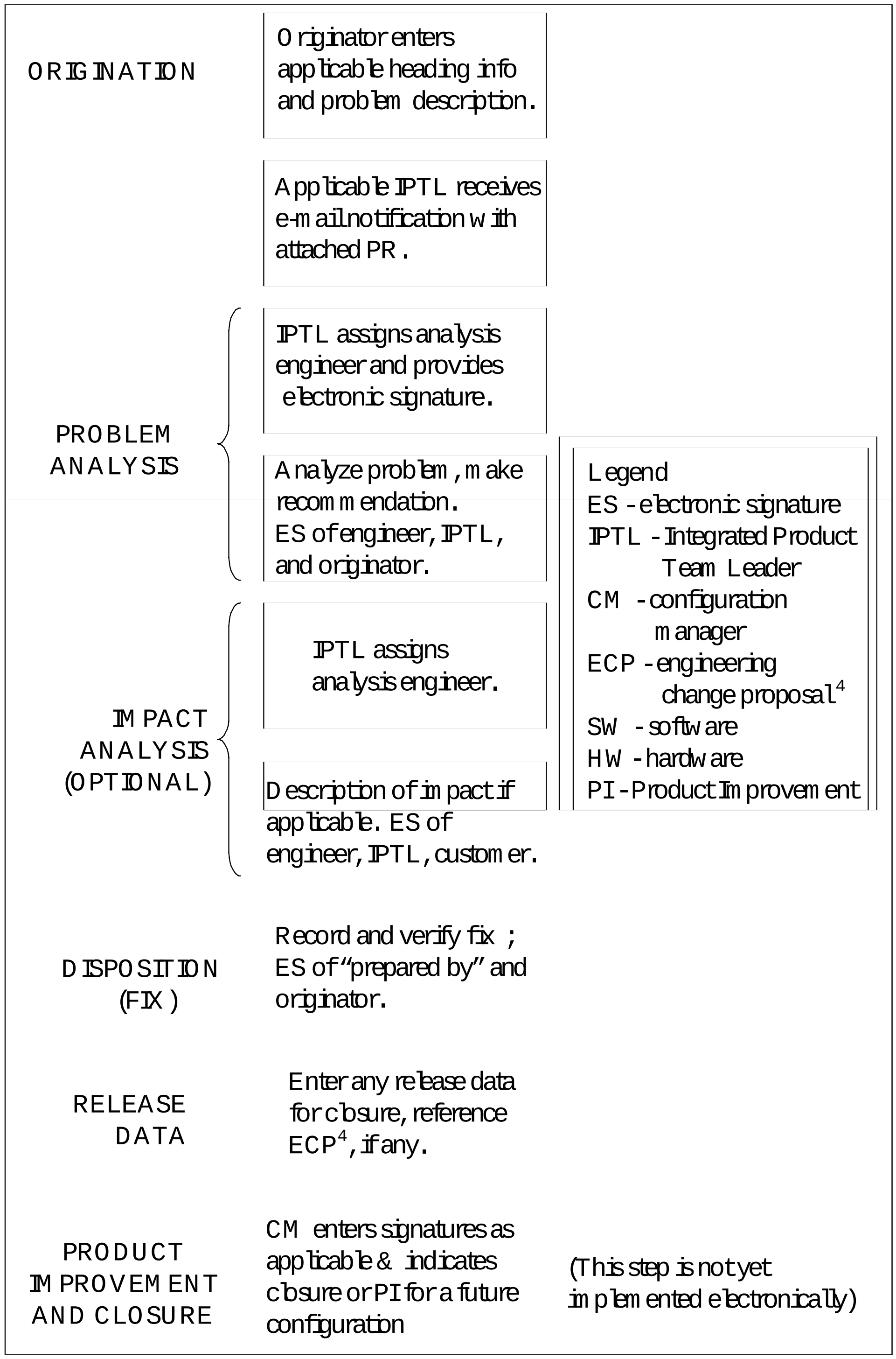

Figure 1. *Boeing Electronic Products (EP) Problem Reporting*

The Seattle system now has approximately 1,000 users with about 40,000 recorded problems. The database tracks and creates reports for any type of problem encountered by EP (product, test, obsolete parts, production) as well as lessons learned, action items, corrective actions, and PRs against the database itself.

Intranet

Access via the Boeing web is partially implemented. This provides access from more locations but with slower response times than those experienced by users who access the system directly using the software application windows front end. The database is migrated off the VAX to a Unix operating system.

Lessons Learned

You may consider commercial off-the-shelf tools that may meet most of your requirements or investigate other organizations or divisions using a tool similar to your requirements. Implementing this system has been a lot of work spread sporadically over nine years. Remember, EP had in-house Oracle developers; many organizations do not have similar resources.

Establish a team to consider new requirements. One result of nine years of experience is EP's use of an Engineering Review Board that helps determine if a change or enhancement will benefit all users. This database has a PR type, which enables the users to write a PR against the PR system. These PRs include problems encountered and enhancements.

Designate a focal point to write and transmit requirements to the database programmer. This should be one or two people who are familiar with the database and who interface well with the programmer. Once approved by the Engineering Review Board, the focal point writes or rewrites the requirements to ensure they are easily understood by the programmer. The change/enhancement is added to the users manual and if the change/enhancement is substantial, training is scheduled.

Consider the importance of your testing approach. For the first four years, a separate test database was maintained. Later, the test database was dropped because of the work required to maintain it. Testing is now performed on the production database. This has been a cost-effective compromise that has resulted in few bugs.

Make sure there is room to grow. In this case, text fields have grown from 20,000 characters to 65,000. Other organizations may decide to use the database once it has proven itself. Also, the more you give users, the more they seem to want.

Sponsorship can make it easier. Once senior management saw the benefit of using a standard process, a single PR, and a single database, EP no longer had to maintain separate requirements, processes, procedures, and updates.

Technology improvements can make it easier. By providing pull-down and pop-up lists within the PR fields, diverse users could live with the generic PR.

Training is essential to the effectiveness of the database and the user. It saves users hours of time by not having to struggle with an unknown element, and gives them hands-on experience.

Help lines and training manuals are important in helping users further understand what they are expected to do. Help lines appear throughout most of the screens to inform and ensure that the PR is completed accurately. The training manual for this database is set up for step-by-step instructions with visual aids.

Summary

Boeing's need for a CM database has been outlined. Defining the database requirements in specific terms for design and implementation presented some challenges. The implementation presented user interface challenges. The system has evolved to meet new requirements and to exploit new technology.◆

About the Author

Susan C. Grosjean is a configuration management specialist with the Electronic Products Group. She has 20 years of configuration management experience at Boeing. Prior to developing the database described in this article, she developed status accounting databases for the B1-B and B2 bombers.

Boeing Commercial Airplane Group
P.O. Box 3707, No. MS OU-47
Seattle, Wash. 98124-2207
425-266-3731
Susan.Grosjean@PSS.Boeing.com

Notes

1. Electrical and electronic devices in aviation, including the software embedded in those devices.
2. For BCAG, the 777 airplane represented an unprecedented software challenge in terms of the size and complexity of the airplane's systems. The 2.5 million lines of newly developed software were approximately six times more than any previous Boeing commercial airplane development program. Including commercial-off-the-shelf and optional software, the total size is more than 4 million lines of code. This software was implemented in 79 different systems produced by various suppliers throughout the world. *CrossTalk* January 1996, "Software Development for the Boeing 777."
3. Line replaceable unit is the terminology for the circuit board, drawer, or cabinet that can be removed and replaced when hardware failure occurs.
4. Class I Changes are called Engineering Change Proposals (ECPs) and consist of changes that are so extensive that additional customer funds must be contracted to implement the change. Changes smaller that Class I are called Class II.