Abstract - The system software for the Consolidated Automated Support System (CASS) contains approximately three thousand modules and nearly two million lines of source code. This software is currently being used on several different configurations of CASS at multiple sites to develop and run numerous Test Program Sets (TPS). Because of the complexity of the software, the vast quantities of code, and the large number of users; proper configuration management (CM) of the CASS system software is crucial to the overall success of the CASS program.

This paper will briefly describe how CASS system software CM is performed. Topics include the requirements of CASS users, configuration identification, configuration status accounting, the change control process, and distribution of software.

I. INTRODUCTION

CASS, the Navy's newest Automatic Test Equipment (ATE), was designed to replace existing United States (US) Navy ATE. This will be accomplished by using several CASS hardware configurations plus ancillary equipment. Currently, CASS is being used for new TPS development, transition of existing TPSs, and Fleet support. This diverse application plus the high number of TPSs required for Unit Under Test (UUT) support will require strict adherence to the principals of CM. Specifically, CM as applied to the system software will be particularly challenging. The source code modules have many interdependencies and relationships which will require strict adherence to change control processes and procedures. The sheer size of the TPS development effort will extend CASS system software CM efforts out over many years in time and many versions of the software. A necessary goal will be the upwards compatibility of all versions of the system software. The large number of CASS stations operating in many environments will provide challenges in software distribution and version control.

The Naval Aviation Depot (NAVAVNDEPOT) North Island (NORIS), as the CASS Cognizant Field Activity (CFA), together with the Naval Air Warfare Center, Aircraft Division, Lakehurst (NAVAIRWARCENACDIVLKE), as a Participating Field Activity (PFA), are responsible for the CASS system software CM efforts.

II. BACKGROUND

A. System Software.

The system software for CASS consists of the software distributed by the government that is used to:

* develop, test, verify, validate, maintain, and execute CASS TPSs
* record station and UUT related maintenance information

Certain commercial software products, such as the digital automated test program generator, are used in the development and maintenance of CASS TPSs; however, those products are not considered CASS system software because they are not distributed by the government.

1) System Software Composition: The two main cateagories of CASS system software are the support software and the station software.

The support software, which is hosted on off-line computer systems, is used for the development and maintenance of TPSs. The support software includes a number of Computer Software Components (CSC), which are mid level software configuration items (CI). Several examples of support software CSCs are the ATLAS compiler, the compiler related processors, the test executive (TE) simulator, and the Master Test Program Set Index generator.

The station software portion of the CASS system software is composed of all of the software that normally resides on the station's fixed disk drive, with the exception of the CASS maintenance software. This includes all of the software used to run TPSs and to record station and UUT related maintenance data. A few examples of station
software CSCs are the operator interface, the TE, the Asset Allocation, and the IEEE 488 Translator.

2) **Quantity of Code:** Between the support software and the station software, there are approximately 1.9 million lines of source code consisting of nearly three thousand Computer Software Units (CSU). A CSU is a single software module, the lowest level software CI.

3) **Complexity of Code:** The CASS system software has a high degree of complexity for several reasons.

First of all, ATE system software is, in general, relatively complex. For example, operating system software, which is an essential component of ATE's system software, is typically one of the most complex types of software in existence. In addition, TE's provide real time command and control of numerous functions on many different instruments to produce and process wide varieties of stimuli and measurements.

The CASS system software is also more complex than that for typical ATE systems because it was designed to be used with multiple hardware configurations. The single set of CASS system software is now used to develop and execute TPSs on all three existing configurations of CASS: the Hybrid; the Radio Frequency (RF); and the Communications, Navigation, and Identification (CNI) stations. After modifications, this software may also be used with the Electro-Optical station currently under development and possibly additional configurations of CASS. Most ATE system software is designed to work only with a single hardware configuration.

A third reason for the CASS system software's complexity is the vast number of interrelationships and interdependances between:

- the CSUs within each CSC. For example, a change to any particular TE module could impact many other TE modules.
- the CSCs within each Computer Software Configuration Item (CSCI), the highest level of software configuration item. For instance, changing the Intermediate Maintenance Operations Management (IMOM) Data Processing software, which is part of the IMOM CSC, may impact other IMOM CSCs, such as the IMOM Pretest software.
- different CSCIs. For example, a change to the ATLAS compiler (part of one CSCI) may impact the TE (part of another CSCI) because the compiler's output must be correctly interpreted and processed by the TE.

B. **CASS System Software Users.**

The CASS project calls for not only the largest planned quantity of testers under any Navy project, it also calls for the largest number of users of any Navy tester. These users can be grouped into two main types:

- users that develop and/or maintain TPSs
- users that execute TPSs

1) **TPS Developers/Maintainers:** Users that develop TPSs utilize both types of the system software (support software and station software) during the development, integration, and testing of TPSs. Similarly, sites that maintain TPSs utilize the same system software to make changes to the existing code and then test the updated software.

Since the use of CASS system software for TPS maintenance software efforts is virtually identical to that for TPS development efforts, further discussions about TPS Developers and their requirements will also apply to TPS Maintainers and their requirements.

All TPS Developer sites are provided the support software as government furnished equipment (GFE). However, relatively few TPS Developers are provided CASS stations and the station software as GFE. The majority of TPS developers utilize the stations and station software at Test Integration Facilities (TIF). TIFs are government managed facilities that have multiple CASS stations, each of which may be used by a number of TPS developers.

2) **TPS Users:** Sites that utilize CASS to execute TPSs use only the station software portion of the system software; they do not use the support software portion. The list of these sites currently includes the Fleet, training facilities, and other Fleet support activities. In the future, this list will also include Foreign Military Sites (FMS) and other United States government agencies such as the National Weather Service.

C. **User Requirements.**

The two types of CASS users, the TPS Developers and the TPS Users, have some common and a few unique CASS system software related requirements.

1) **Common:** All CASS users share a number of system software related requirements, including the following:

- an effective means for reporting problems and proposing enhancements to the software
- periodic updates to the software to implement
enhancements and correct problems

- assurance that each site is delivered only those software versions that are authorized to be used by the site

2) TPS User Unique: The sites that execute TPSs require that every new baseline of the station software run every existing TPS at least as well as any previous baseline of the station software. This is required for several reasons. Loading station software, for example, takes time away from using the station to run TPSs and, therefore, should be avoided. Also, TPS user sites can manage the CASS system software far easier if only one version of the software is used and stored at those sites.

3) TPS Developer Unique: TPS Developers have several unique requirements of CASS system software.

One unique requirement CASS TPS Developers have of system software deals with the speed of the problem reporting process. In the world of TPS Development, time is money. When a CASS anomaly impacts TPS development, the time spent forwarding reports and related information between activities/companies can be costly. In order to minimize these costs, TPS Developers require problem reporting and tracking methods that are timely as well as effective.

Another unique requirement TPS Developers have of system software deals with rapid changes to the software. TPS Developers require timely responses to anomalies that impact TPS development efforts. In accordance with TPS contracts generated using the Red Team Package, the government response is required within 30, 60, or 90 days, depending on the severity of the anomaly. Under certain conditions, the proper response to an anomaly report means that a system software change must be developed, integrated into the rest of the software, and then delivered to the TPS Developer within the proper response time.

TPS Developers also require that all new versions of the system software have minimal negative impacts to TPS development efforts. For example, a new version of system software that is incompatible with a TPS under development could increase the cost and delay the delivery of that TPS.

A fourth TPS Developer related requirement deals with the management of multiple versions of system software used at a site. Any particular TPS development site may be using several versions of system software at any one time. This is possible because the TPSs being developed at a site may be associated with a number of contracts, each specifying a different version of system software.

TIFs, for example, manage stations being used by multiple TPS Developers working on numerous TPSs. In addition, unique versions of system software, generated in response to anomaly reports, are also used by TPS Developers. In order to adequately manage all versions of the software, TPS development sites require that all updates to the system software be both easily identifiable and controllable.

D. Configuration Management.

The method used to meet these challenging requirements involves careful management of the configurations of the system software. This type of management, which is called Configuration Management (CM), can be defined as the management discipline applied to CIs over their life cycle to ensure that the characteristics of CIs meet defined user requirements.

According to [1], CM for all CIs should address the following four elements:

- configuration identification
- configuration control
- configuration status accounting (CSA)
- configuration audits

All CIs require some degree of CM. However, the amount and the intensity of CM efforts vary from project to project, depending on the complexity of and the demands on the CIs. For example, a simple CI used in a noncritical application by a single user needs relatively little CM discipline. Increasing the complexity of the CI or the user's requirements of the CI normally increases the CM requirements.

E. Previous CASS CM Responsibilities.

CASS CM responsibilities were delegated to the prime contractor by the Naval Air Systems Command (NAVAIRSYSCOM) Program Manager (PMA-260) during the development phase of CASS CIs. During this period of time, the government's CM role was limited to monitoring contractor performance and providing program oversight.

The development phase for the three existing CASS configurations and related software ended in November 1993 when the provisional product baseline (PBL) was established and government assumed ownership of these CIs.

F. Present CASS System Software CM Responsibilities.

According to [2], all CM responsibilities for the existing
CASS CIs were resumed by the government upon establishment of the provisional PBL. At that time, NAVAIRSYSCOM assigned NAVAIRNSDEPOT NORIS responsibility for the existing configurations of CASS. In addition, NAVAIRWRCNACDIVLKE was assigned as the PFA for CASS system software.

G. Past CM Experiences.

Both NAVAIRWRCNACDIVLKE and NAVAIRNSDEPOT NORIS have significant experience in CM of ATE CIs. NAVAIRWRCNACDIVLKE, for example, has been responsible for CM of the majority of the Navy's common ATE system software for over ten years. Likewise, NAVAIRNSDEPOT NORIS has been responsible for CM of the Navy's common ATE hardware during that same period of time.

III. CASS SYSTEM SOFTWARE CM

In spite of the magnitude, complexity, and user's unique requirements of the CASS system software, standard CM techniques are being used to provide the users with quality system software products. The following illustrates the current CASS CM organization and briefly describes the four elements of CM (Configuration Identification, Configuration Control, CSA, and Configuration Audits) for the CASS system software project.

A. Current CM Organization.

The organizational relationships of the groups/activities involved in CASS CM since November 1994 are thoroughly described in [2]. The relationships involved with CASS system software CM, a subset of those described in [2], are depicted in Fig. 1 and are briefly discussed in the following paragraphs.

1) ATE Program Manager: PMA-260 provides broad program direction and resources to the CASS Program Manager.

2) CASS Program Manager: PMA-260D3 provides specific program direction to the CASS Technical Team Leader as to the conduct of the entire CASS project.

3) CASS Technical Team Leader: PMA-260D31 provides working level program management, technical advice, and guidance to the CFA.

4) CASS CFA: NAVAVNSDEPOT NORIS performs, among other things, CM functions as the CASS Configuration Manager.

5) CASS System Software PFA: NAVAIRWRCNADLKE performs In-Service Engineering (ISE) and system software CM work delegated by the CFA.

6) Others: There are several other groups/activities involved, to a somewhat lesser extent, with the CASS system software CM.

The CASS Technical Working Group (TWG) provides technical support for various aspects of the CASS project. This group is lead by PMA-260D31 and is composed of representatives from the CFA, PFAs, the Naval Weapon Support Center at Crane, and the Naval Air Warfare Center Aircraft Division (NAVAIRWRCNACDIV) Patuxent River (PAX).

NAVAIRWRCNACDIV PAX contributes to the CASS CM efforts by providing testing, validation, and verification of new versions of software.

Finally, there are software change review boards (SCRB) at both NAVAVNSDEPOT NORIS and NAVAIRWRCNACDIVLKE as well as the PMA-260 support equipment (SE) software change control board (CCB). As described in [2], the two SCRBs review change control documentation prior to submission to the PMA-260 SE software CCB for approval.

Fig. 1 Organization Chart
B. Configuration Identification.

According to [1], configuration identification includes:

* the selection of Cls
* the determination of the types of configuration documentation required for each Cl
* the issuance of numbers and other identifiers

All of the CASS system software configuration identification related work required for the establishment of the CASS PBL was completed prior to the completion of the Physical Configuration Audit (PCA).

After the PBL was established, additional configuration identification work was required in order to assist users in identifying certain types of software. Table 1 lists a few examples of these new assignments.

TABLE 1
Newly Identified Cls

<table>
<thead>
<tr>
<th>PART/VERSION NUMBERS*</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASS-1-XXX</td>
<td>All software on the CASS station disk that contains the control software, Digital Test Unit software, Self-Maintenance software, etc.</td>
</tr>
<tr>
<td>CASS-2-XXX</td>
<td>All software on the CASS station disk that contains the CASS station operating system and computer diagnostics software.</td>
</tr>
<tr>
<td>CASS-SUP-XXX</td>
<td>All CASS support software issued by the Government and used on off-line computer systems to develop CASS TPSs. Includes the Test Program Set Development Shell, ATLAS Compiler, etc.</td>
</tr>
<tr>
<td>XX.X</td>
<td>ATLAS Language Processor Software (ALPS). Includes the ATLAS Compiler, Binder, Look-up Table, etc.</td>
</tr>
<tr>
<td>CASS-1-XXXEX</td>
<td>Engineering Release of the CASS-1-XXX software.</td>
</tr>
<tr>
<td>CASS-1-XXX.X</td>
<td>Interim Release of the CASS-1-XXX software.</td>
</tr>
</tbody>
</table>

* The "X"s represent version numbers.

Configuration control of baselined software involves the following elements:

* effective means, as applicable, for proposing changes to the CSCIs
* effective control of all CSCIs and their approved configuration documentation
* implementation of approved changes

1) Effective Means for Proposing Changes: There are currently a number of different methods for initiating proposed changes to the CASS Cls. Having more than one method may appear to be redundant but is actually necessary due to the variety and special requirements of CASS users. Methods for initiating proposed changes include the use of the following types of proposed enhancement and problem reports:

* Pre-Planned Product Improvements (P3I) - used to identify proposed enhancements to CASS. P3Is are typically generated by the TWG as a result of investigations into changes in threat that require increased capability and investigations of technology breakthroughs.

* Problem Identification Reports (PIR) - the prime contractor's in-house document used to identify CASS anomalies.

* System Problem Reports (SPR) - used to identify and report CASS anomalies that impact TPS development efforts. SPRs are normally submitted electronically by TPS Developers using CASSTRAC in accordance with the Red Team Package. CASSTRAC, a menu-driven relational database hosted on a central computer, is used to both submit and process SPRs in a timely manner.

* Quality Deficiency Reports (QDR) and Engineering Investigation (EI) - according to [2], the standard support equipment (SE) problem reporting formats used by all US Navy Fleet sites at the organization and intermediate levels of maintenance. These formats are also used by FMS customers to report problems.

Whenever the investigation of a problem report determines
that a change to a CASS CI is required, a proposal to change the CI is generated. The three types of proposals to change CASS system software are the Rapid Action Solution (RAS), Software Change Request (SCR), and Software Change Proposal (SCP).

The RAS and SCR are used to identify and track proposed changes related to a single problem. RASs are used to document SPR related changes. SCRs are primarily used to document and track changes originating from all other types of problem reports, such as EI requests. However, to simplify the tracking of changes, all approved RASs are eventually assigned SCR numbers so that SCR numbers alone can be used to track all proposed changes to the software baseline.

The SCP is a form used by the government to propose and effect software changes to the current approved baseline. To minimize the overall costs to the CASS project, SCPs for system software normally address batched SCRs. However, a SCP may address a single, relatively complex and/or expensive proposed software change.

2) Effective Control: The effective control of the CASS system software and related documentation can be divided into three areas: controlling access to the source code, controlling changes to the software baseline, and controlling distribution of the executable code.

Controlling access to the software source code is crucial in order to avoid bootleg and tainted versions of the software. The source code is protected from tampering and unauthorized duplication in several ways. For example, the computer systems hosting the source code reside in a controlled access room, as described in [2]. The code is further protected from tampering by limiting the number of computer accounts that have read and write privileges and severely limiting distribution of the source code. In addition, all backup copies of the source code are stored in secure locations.

Controlling changes to the system software involves:

* the regulation of the flow of proposed changes
* documentation of the complete impact of the proposed changes
* review of the proposed changes
* approval of the proposed changes

The change control process for CASS system software is described in detail in [2] and is depicted in Fig. 2.

Software engineers investigate all system software related anomalies/problems and, when applicable, propose a change to the system software baseline. Proposed changes that are successfully tested and approved will typically be batched together, incorporated into the existing software baseline to create a new baseline, and then forwarded to all CASS users as part of a "periodic release" of the system software. All proposed changes also describe impacts, if any, to documentation.

Each proposed change is reviewed by the two SCRBS and the TWG for technical and business merit, as stated in [2]. Proposed changes costing more than a specified dollar value are also reviewed by PMA-260.

When necessary, temporary workaround procedures (TWPs) and temporary solutions are issued to selected sites until a permanent solution has been processed, tested, implemented, and released as part of a periodic release. TWPs and temporary solutions are necessary to respond to the problems reported by the users in a timely manner. TWPs may be issued to all CASS sites, but temporary solutions are issued only to TPS Developer sites.

Temporary solutions are versions of the CASS system software used during TPS development. The two types of temporary solutions are called "engineering release" and "interim release". An engineering release contains a

![Fig. 2 CASS Software Change Process](image-url)

---

12
proposed solution to a single (normally high priority) SPR and is typically only sent to the TPS development site that submitted the SPR. An interim release contains the proposed solutions to more than one SPR and is typically sent to multiple TPS development sites. Engineering and interim releases are primarily issued in order to meet the government's 30-60-90 day response to SPRs as described in the Red Team Package. Temporary solutions are assigned unique software version numbers and are issued as complete software packages rather than as patches in order to assist TPS development sites managing multiple versions of system software.

After a proposed change is generated, the associated code is subject to an appropriate amount of testing. The goal of this testing is to verify that a proposed CASS system software change resolves the problem(s) that it was designed to resolve and verify that the change will be upwards compatible, or will not cause any other problem.

In theory, it may be possible to guarantee that any given software change will not cause other problems. However, due to time constraints and the complexity of the CASS system software, it is practically impossible to guarantee upwards compatibility for all releases of CASS system software. Therefore, all periodic releases of system software are subjected to extensive testing prior to their final approval and release to greatly reduce the likelihood of impacts to Fleet support and TPS development efforts. The software baseline established upon SCP approval contains all of the approved changes associated with that SCP.

The implementation of approved changes involves both the implementation of the individual changes into the software baseline and the installation of the updated software at CASS sites. The implementation of the approved changes into the system software baseline is an integral part of the change control process briefly described in the previous section. The reviews of the SCP and the testing of the software ensure that all approved changes scheduled for the periodic release have been implemented and that unapproved changes are not implemented. The software baseline established upon SCP approval contains all of the approved changes associated with that SCP.

The distribution of the software is controlled in order to prevent sites from being sent versions of software that they are not authorized to use. For example, interim and engineering releases are not forwarded to US Navy Fleet sites because Fleet sites are only authorized to use updated versions of software released under a TD.

All software releases are adequately labeled to identify the version and type (e.g. "interim release").

3) Implementation of Approved Changes: The implementation of the approved system software changes involves both the implementation of the individual changes into the software baseline and the installation of the updated software at CASS sites.

The installation of the updated software at CASS sites is addressed by the TD for US Navy sites and by designated government acceptance representatives (DGAR) or their equivalents at most TPS development sites. The TD mandates that each US Navy site install new station software within a specified time period. DGARs and their equivalents typically coordinate with TPS developers to install and use new versions of system software. TIFs install new versions of station software as required by their customers.

D. Configuration Status Accounting (CSA).

CSA can be defined as the recording and reporting of information needed to manage configuration effectively.

For CASS system software CM, there are five important elements of CSA.

The first important CSA element is identifying the current approved configuration documentation and identification number associated with each system software related CI. This is accomplished by the generation of a version description document (VDD) for each software release that lists the software and version numbers associated with that release. Complete VDDs are not normally released to users; however, abbreviated VDDs may be forwarded to some sites.

Another element involves recording and reporting the status of proposed engineering changes from initiation to final approval. This is achieved by tracking and reporting every submitted problem report and proposed software changes.
change from their receipt or generation throughout the remainder of the CASS life cycle. For example, the status of SPRs is recorded and reported via CASSTRAC. Also, as described in [2], all ODRs and EIIs are tracked via an automated on-line CFA ISE information system. Access to this information system is provided real time via modem to all sites requiring status. In addition, a SCR database tracks the status of proposed changes initiated by all types of problem reports, SPRs as well as EIIs.

The third CASS system software CSA element is recording and reporting implementation status of authorized changes. Implementation of each change is recorded using a commercial software management tool when the change is incorporated into the software. Reporting of change implementation is attained by the documentation that accompanies the software release.

The another element is providing traceability of all changes from the original baselined configuration documentation of each CI. Traceability is accomplished by inputting sufficient data into the records of the commercial software management tool.

The last element is reporting the installation status of configuration changes to all system software at all locations. As stated in [2], this is achieved by providing access to a database that contains up-to-date information about the versions of system software installed at each user's site.

E. Configuration Audits.

All audits required for the establishment of the CASS PDL, namely the Functional Configuration Audit (FCA) and the PCA, have been completed for existing systems. If any new CASS configurations requires modifications to the existing system software, then incremental FCAs and PCAs will be performed.

Additional system software audits may be conducted to verify or re-verify the accuracy of the data in:

- the status accounting system
- the configuration change process
- the current product configuration identification
- and the change reporting process.

Any additional audits will conform to [1] and will be administered in accordance with appropriate NAVAIRSYS COM instructions.

IV. CONCLUSION

CASS, with its many configurations and broad application, challenges the CM discipline. Through the application of sound CM principles and attention to the unique requirements of the CASS project, users will continue to be provided with quality system software products that support their needs while maintaining the flexibility to meet new requirements and to resolve identified problems.

REFERENCES