

ON-LINE CONFIGURATION MANAGEMENT

Wynn Gaylor and Ronald Murphy

Gould Inc., Simulation Systems Division
Melville, New York

ABSTRACT

ON-LINE CONFIGURATION MANAGEMENT: Configuration Management is covered in many books and articles and is described in detail for development and test of systems which are either all software or all hardware. In real time simulation which involves integrated and, many times, interactive hardware and software, plus human interaction with instructor and/or students, a new level of control is required called hardware software integration/test. In addition, subjective customer evaluation and acceptance may extend this period indefinitely. This period of test and evaluation could result in elapsed time of half the project schedule and is usually where all the budgets/schedules go down the tubes. This paper describes an approach to on-line system configuration management, starting with a prototype baseline system ready for hardware software integration.

INTRODUCTION

An Industry/Government meeting held on November 26, 1979 prior to the first interservice/industry training conference concluded with a list of initial planning and acquisition issues for a panel discussion during the conference. The purpose was to gain a better understanding of problem areas, to identify and discuss probable causes, and to reach some agreement on possible recommendation to improve performance. Under the topic "Contractor Performance," the following problem areas were described:

"The need exists for the development of more definitive software management plans with milestones adequate for monitoring developmental progress. Concomitantly, determinative plans for the integration of software with hardware is required. Frequently either software development lags and/or extensive integration problems are encountered; the resultant effort to catch up or correct causes delays and additional costs."

"Practically every major training device program has experienced difficulty in the development of software and its integration with hardware. Underestimating software development adversely impacts time, schedule, and performance requirements, and creates serious life cycle support problems. What can be done by contractor management to place the required emphasis on software development?"

"Status reporting systems are only creditable if they factually state progress on the program and highlight critical issues in management or engineering that require decisions. What can be done to improve the quality and timeliness of status information provided by contractors?"

This paper addresses these problem areas, particularly the last paragraph on reporting systems, as they occurred during the development of the T-44 Operational Flight Trainer (OFT). The T-44 OFT development program was the first project which applied the on-line configuration management technique.

DEFINITIONS

Reference Figure 1 for pictorial presentation.

1. On-Line - The complete hardware/software system under operation and test in real time.
2. Verification - Test the computer program to ensure that it satisfies all performance and design requirements; e.g., software engineering tests.
3. Validation - Test the computer program to ensure that it operates properly in the system environment and that it satisfies all system requirements; e.g., system engineering tests.
4. Hardware Software System Integration - This type of testing executes the computer program in a completely real (on-line) environment utilizing actual hardware. This test demonstrates that the computer program satisfies all requirements of the program performance specification while being executed in a real system environment.
5. Acceptance Test Procedures (ATP) or Certification - This type of testing makes use of the computer program after it has completed system hardware software integration. This test is intended to exercise all subprograms of the computer program in consonance with the hardware. These tests certify that all software, hardware, and system requirements have been satisfied. These tests are defined in the Acceptance Test Procedures document. Figure 1 shows the relationship of verification, validation, and certification testing.

CONFIGURATION MANAGEMENT

Configuration management (CM) is a discipline traditionally applied to the development of hardware systems or to the development of hardware elements of hardware software systems. Its application in connection with other disciplines leads to orderly and structured system development. CM is generally concerned with the consistent labeling, tracking, and change control of the hardware elements of a system. Software configuration manage-

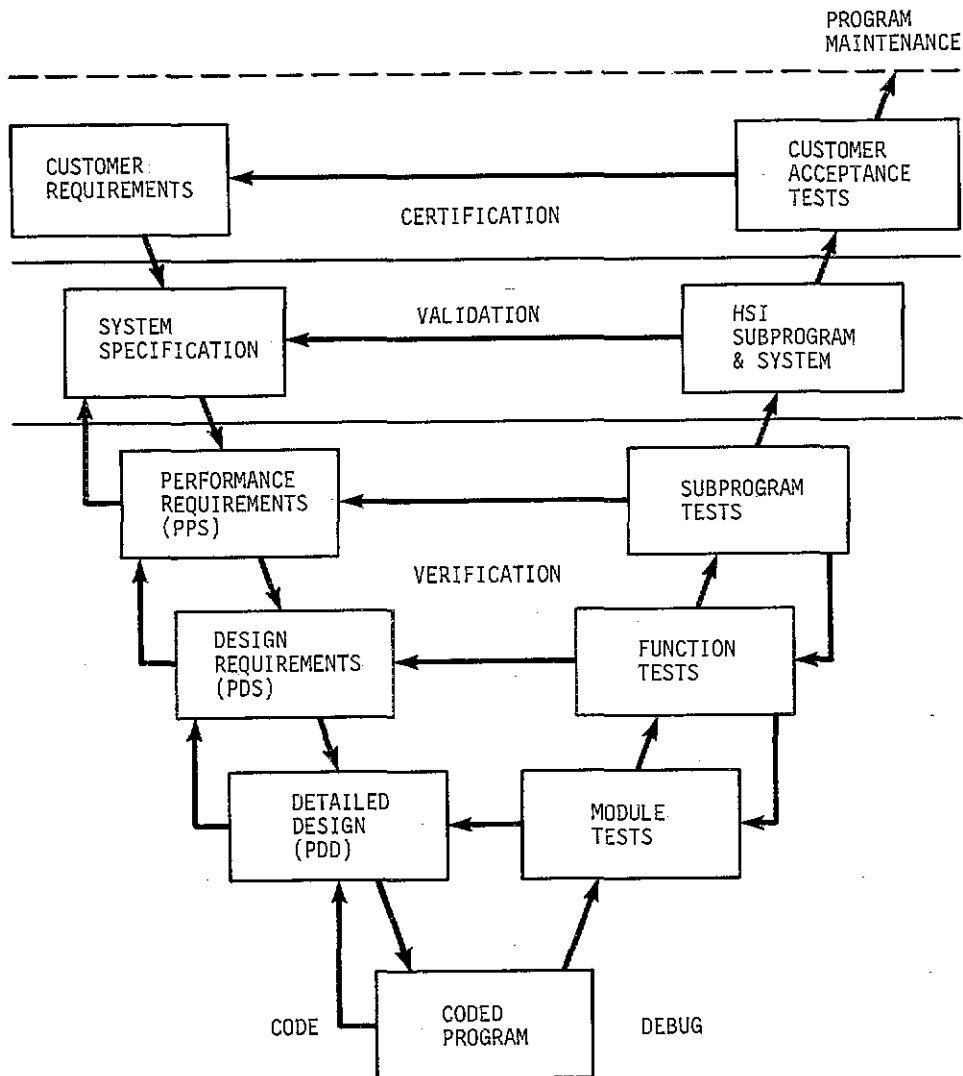


Figure 1. Verification, Validation, and Certification

ment is CM tailored to a system, or portion of a system, predominantly comprising software or firmware.

Figure 2 depicts the stages of the system life cycle. Although each stage is equally applicable to hardware and software, the differences between hardware and software development and production give rise to different specific requirements for managing the software life cycle. Each stage of development culminates in a baseline, (Figure 3) a point at which management has the opportunity to view the system in detail in order to examine its integrity - that is, its adherence to and satisfaction of the operational requirements.

After both hardware and software become independently operational, a period of hardware software integration (HSI) begins and continues until all acceptance testing is completed and the device validated. The basic four components of configuration management will be addressed during HSI and acceptance testing as depicted in Figure 4.

Configuration Identification

Identification during hardware software integration and acceptance testing is well defined and takes place at the subprogram level. A subprogram is a subdivision of a computer program that accomplishes an entire set of

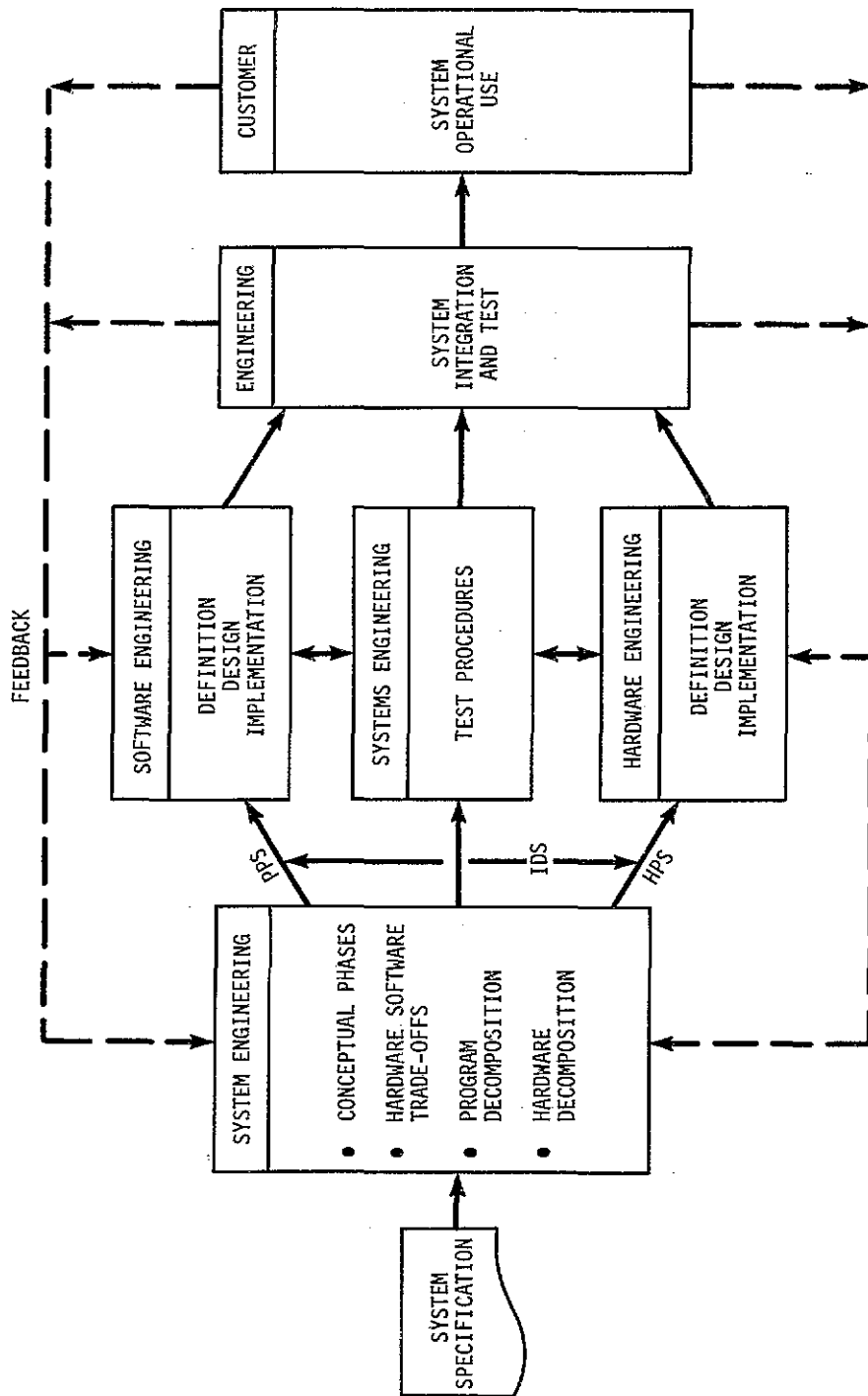


Figure 2. System Life Cycle

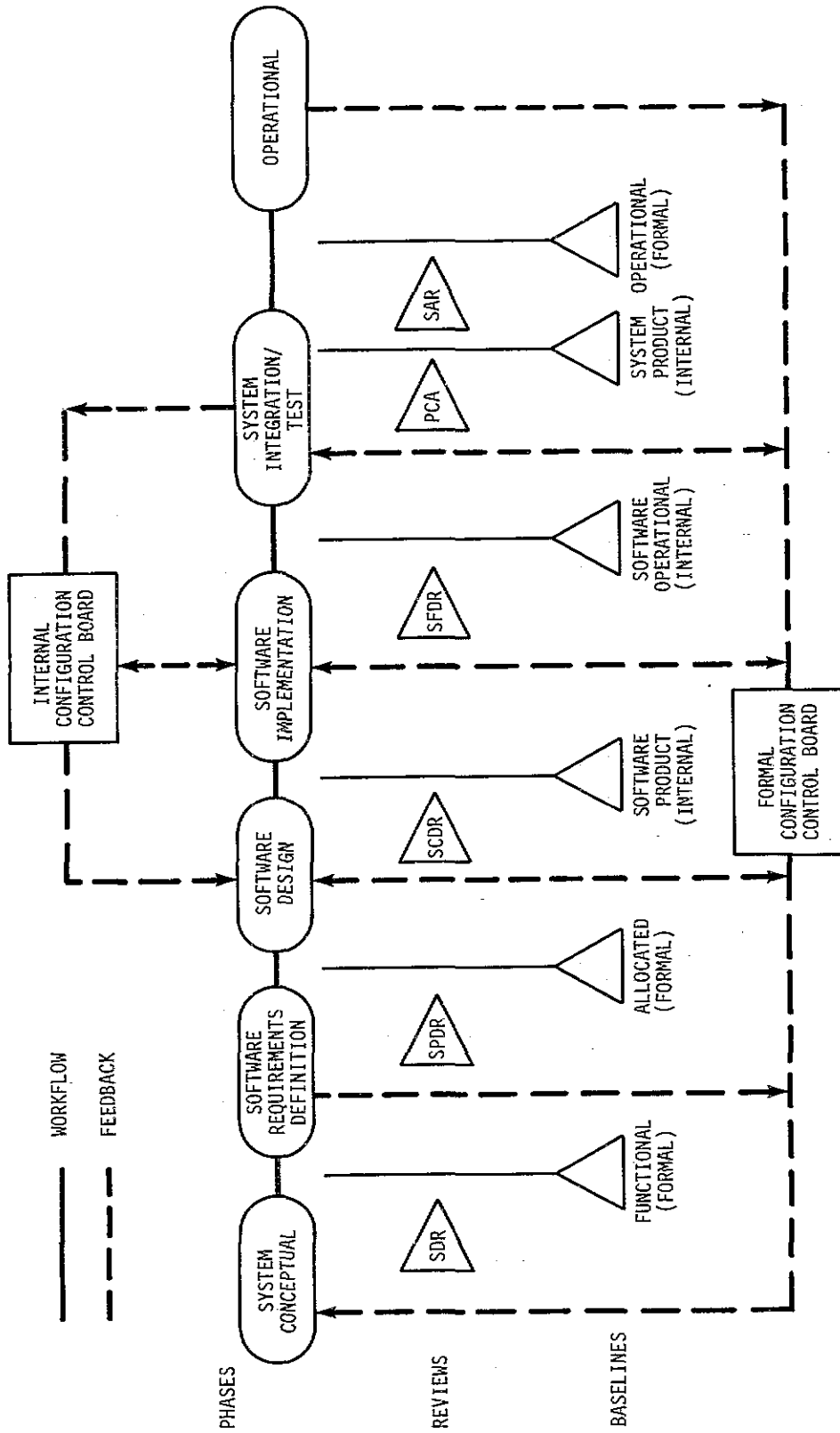


Figure 3. Computer Software Life Cycle

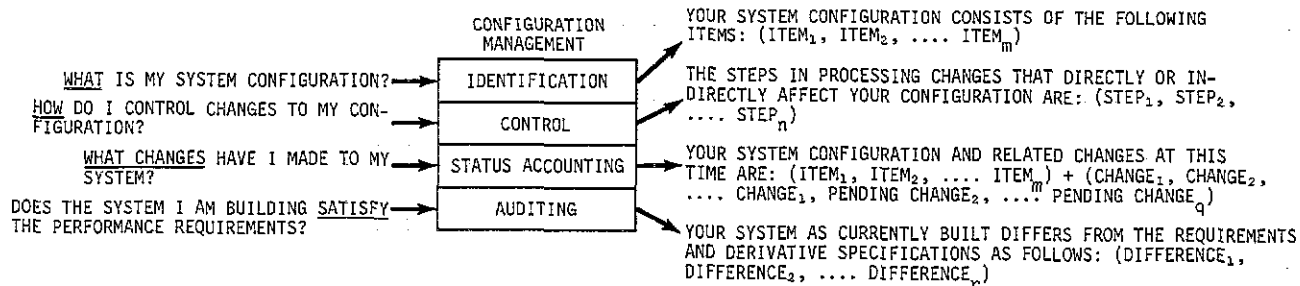


Figure 4. The Four Components of Configuration Management

program performance requirements; e.g., fuel system. Therefore, it is a specific testable entity.

The tests are also well defined. A complete and approved Acceptance Test Procedure is utilized (and proofed) during this period of configuration management. Note that during all the testing we are asking the question, does the system satisfy the performance requirements? The whole testing philosophy is keyed to the audit concept of discrepancies; e.g. "The current system differs from the requirement as follows!" The audit is continuous. The schedule is keyed to the audit.

On the T-44 OFT, the schedule was developed by the project engineer and based on the estimates to complete obtained from his engineering staff, the status of the hardware, and the subprogram interdependencies. Each subprogram was identified in accordance with the approved acceptance test document and organized in schedule format for each of two shifts. Included on the schedule was a weekly summary status accounting of discrepancy reports (DRs). Figure 5 indicates graphically normal DR generation and abnormal DR generation. Note that a normal curve peaks out and drops off towards zero. A glance at the schedule (Figures 6, 7, 8, 9) provides the observer with a top level audit and trend. For example "Are more DR's being generated than fixed?" Having established what the configuration is and a method of determination of status, we will now look into methods of control.

Configuration Control

Controls take two significant forms; the first is control of changes to the software configuration and the second is control of the status of testing. On the T-44 OFT the problem was compounded somewhat by having two devices (trainers) entering HSI at approximately the same period in time. Therefore the following management procedure was established.

Software Configuration Control. Every Friday morning prior to the 0800 shift, system 1 was

backed up on magnetic tape. System 2 was made identical to system 1 by restoring system 1 backup tape.

System 1 was considered the master system. All other systems were considered as test beds for changes. It was the responsibility of each software engineer to update system 1 with any modified models or programs by informing the librarian of the files to be moved to system 1 and on which system they currently resided.

All system discrepancies (hardware and software) were to be reported using the Discrepancy Report (DR) form (Figure 10). The filled in form was given to the librarian who was responsible for assigning a DR number and entering it in the DR master log. Hardware DR's were then passed on to systems personnel for handling. As each DR was corrected, an entry to that effect was made in the master log. The DR was then defined as "TESTED". A DR could only be defined as "CLOSED" when tested and signed off by the project engineer.

As each DR was tested and closed, the backup tape number was recorded to provide traceability for all changes. All changes were in the form of source code updates. No patches!

A job accounting system incorporated into the vendor's computer operating system was used as a general accounting ledger to track all changes. The accounting file was listed every Friday morning and also saved on the system backup tape. After the file had been saved, it was purged and made ready for the next week. The job accounting listing, the backup tape number, and the DR masterlog provided an excellent means of managing the progress of the project and controlling the configuration of the system.

Testing Status Control. System test control was established by setting up good communications and by test status documentation.

Communication:

Established weekly all hands meeting.
Wrote minutes for review.
Determined significant problems.
Established responsibilities and action plans.

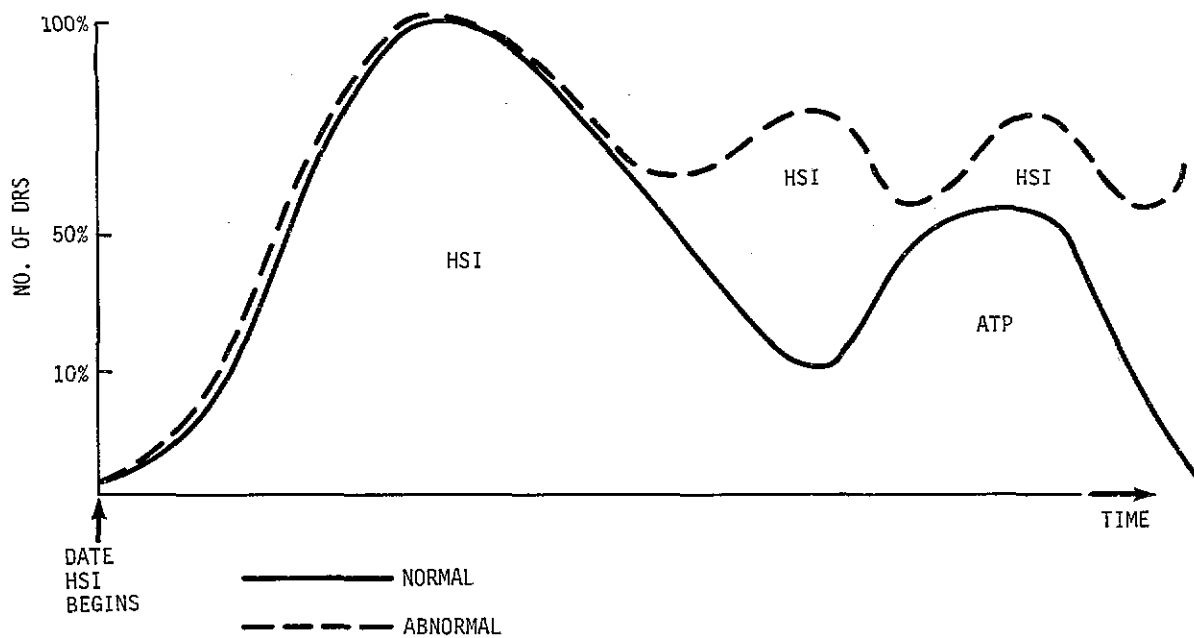


Figure 5. Normal/Abnormal DR Generation

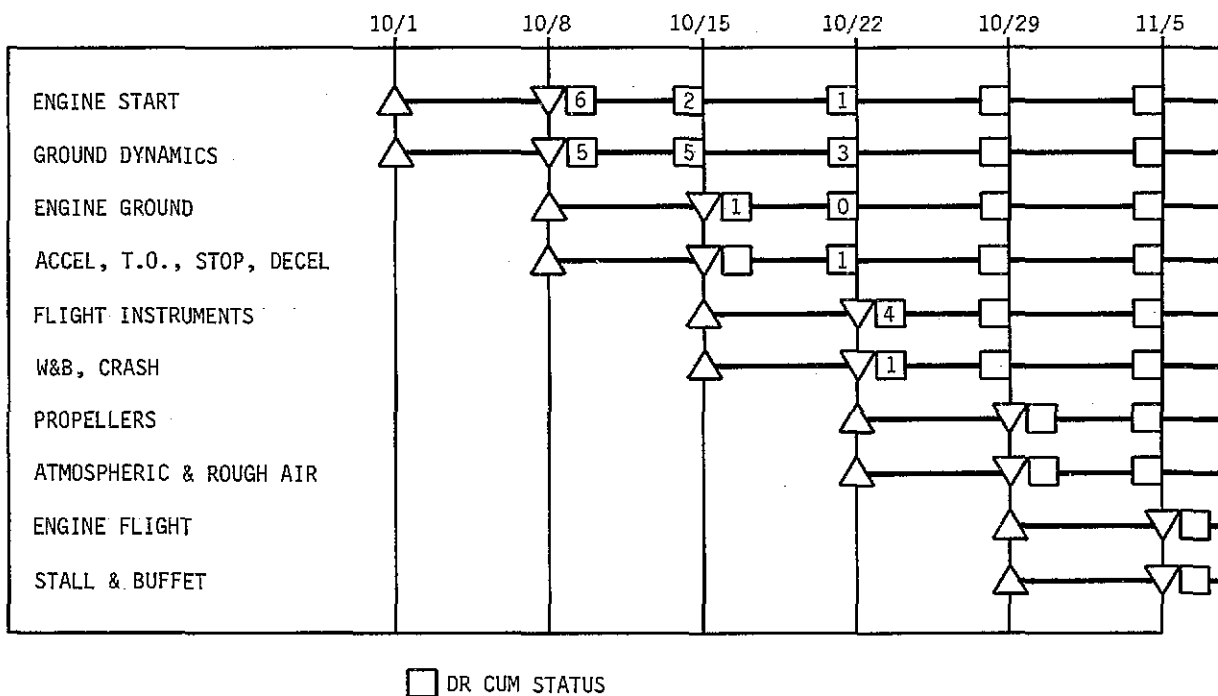
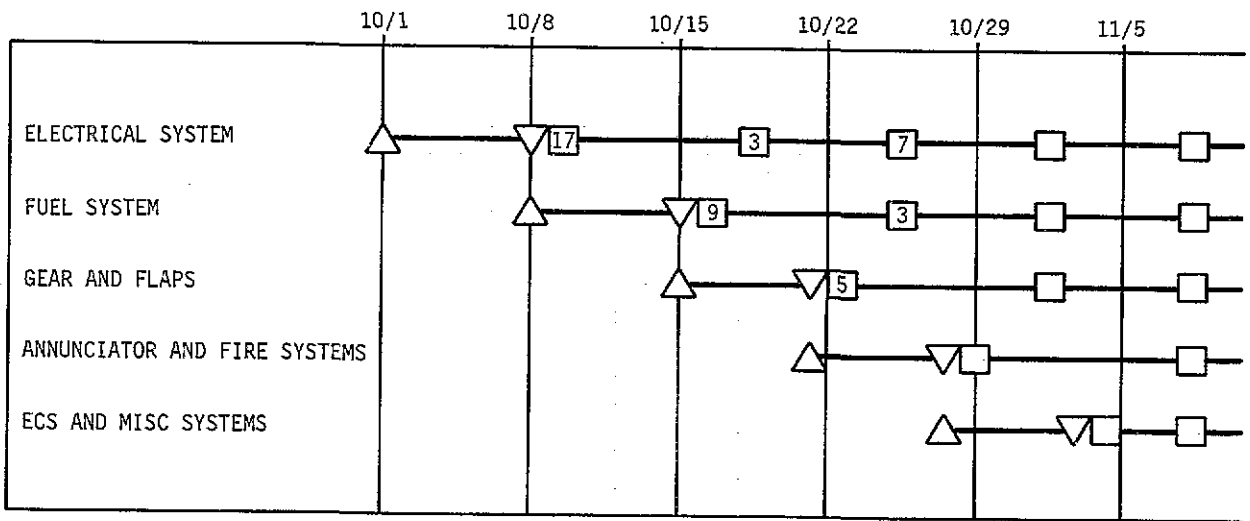
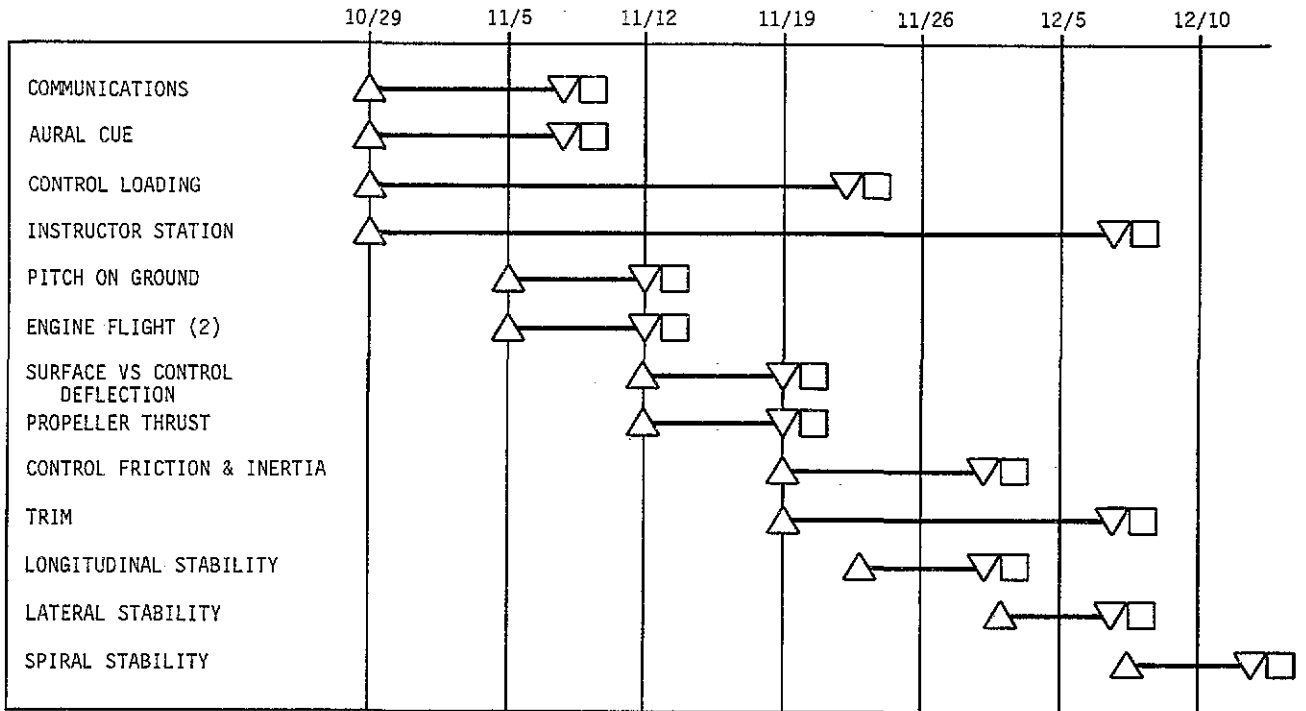


Figure 6. T-44 System Integration and Evaluation Shift 1 (Month 1)



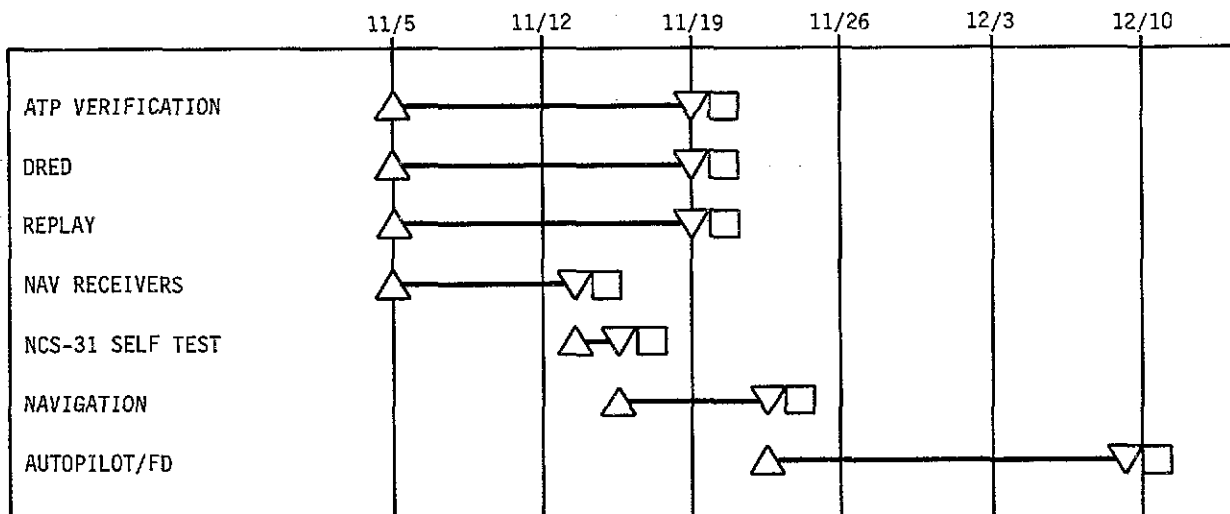
□ DR CUM STATUS

Figure 7. T-44 System Integration and Evaluation Schedule Shift 2 (Month 1)



□ DR CUM STATUS

Figure 8. T-44 System Integration and Evaluation Schedule Shift 1 (Month 2)



DR CUM STATUS

Figure 9. T-44 System Integration and Evaluation Shift 2 (Month 2)

DISCREPANCY REPORT (DR)

PROJECT: _____ DATE: _____

ISSUER: _____ PROBLEM NO.: _____

RESP. ENG.: _____ REF: _____

TYPE OF PROBLEM

() HARDWARE: SYSTEM # _____ () SOFTWARE: SYSTEM _____

STATEMENT OF PROBLEM

CORRECTIVE ACTION TAKEN

TESTED

DATE _____ TAPE # _____ ENGINEER _____

DATE _____ TAPE # _____ ENGINEER _____

COMMENTS

Figure 10. Discrepancy Report Form

Accounting:

- Set up test schedule keyed to discrepancies.
- Updated discrepancy form.
- Librarian had full authority and responsibility for accounting.
- Job accounting system incorporated into the operating system.

The status accounting report, described below, was utilized at the weekly meetings and proved to be the key ingredient to successful communication. Each person knew his responsibility and management visibility allowed the decision-making processes to work!

Status Accounting

In order to keep track of all hardware and software problems which occur during Hardware Software Integration (HSI) or the Acceptance Testing Procedure (ATP), a Discrepancy Report (DR) is filled out. As each discrepancy is reported, a control number is assigned by the librarian and entered in the master log. The function of this log is to keep a record of all DRs and present this information in various ways.

The DR Status Accounting Report, written in FORTRAN IV, will operate on the project computer system in the interactive mode. This interactive program reads a master log entry in card image format, processes the input data, and prints a hardcopy output report as specified below.

The program can be applied to any trainer project by preparing an appropriate set of inputs. The program input file (project data base) can be divided into two sections. The first part (systems section) describes the major systems of the trainer and generally correlates with the computer subprograms. The second part (data section) of the

file represents the DR data base. It is information that is taken from the DR log and added to the data base periodically. Thus the systems data remains fixed for a given project while the DR data base increases as the project progresses.

The first record of the systems section contains the name of the program (i.e., T-44, C-130, 21E17, etc.). Next are the system descriptions, one per record, along with a unique system letter. After the system description comes the data section. This information includes the DR number, system letter(s), initials of the assigned engineer, date tested (when applicable), failed testing or waiting for parts indicators, date closed (when applicable), a brief description of the problem, and the number of the backup tape on which the problem solution first appears.

The program reads the input file and prepares the master DR log which contains the status of each DR printed out in numerical order. A summary table is compiled which lists each system, the number of open, closed, or tested DRs, the total number for each system, and the totals for the whole job.

The information is then resorted. The first resorting procedure prints out all DRs which are not closed. The next sort will print out a listing of all DR's which are ready for testing. The non-closed DRs are then resorted again for a third and fourth time and printouts are made by system and then by assigned engineer.

Last, another table is compiled showing the DR totals - open, tested, closed, and totals - by responsible engineer.

Figure 11 identifies the format and contents (as described above) and indicates that the loop is closed by summarizing the DR's by subprogram on the schedule.

1. FORMAT:	DR. NUMBER	SYSTEM AFFECTED	RESP ENG	DATE TESTED	DATE CLOSED	DESCRIPTION	STATUS	TAPE NO.
2. CONTENTS:	<ul style="list-style-type: none">● SYSTEM LEGEND - IDENTIFICATION● MASTER DISCREPANCY LOG<ul style="list-style-type: none">- SYSTEM SUMMARY STATISTICS● OPEN DISCREPANCY LOG<ul style="list-style-type: none">- READY FOR CLOSEOUT TEST● STATUS OF EACH DEFINED SYSTEM● RESPONSIBILITY LOG<ul style="list-style-type: none">- INDIVIDUAL RESPONSIBILITY STATUS							
3. UPDATE:	<ul style="list-style-type: none">● DAILY● WEEKLY - DISTRIBUTED AND REVIEWED							
4. SCHEDULE:	<ul style="list-style-type: none">● SETUP LOGICAL SYSTEM TEST SEQUENCE● KEYED SCHEDULE STATUS TO DR'S● TRACK DR HISTORY							

Figure 11. T-44 Job Accounting

Auditing

The final audit process follows naturally from the HSI in that system testing proceeds into Government acceptance testing utilizing the Government-approved document. This occurs after in-house testing has reduced all significant discrepancies to zero.

CONCLUSIONS

The approach described herein provided never-before-attained management visibility into audit status. All parties were able to review detailed printouts as to the exact status of the project. Responsibilities were clear and it was obvious that the job was under control. The printouts proved to be extremely valuable when used as the basis for communications, both at in-house meetings and for the customer's review.

As a result of this experience, Gould, Inc., Simulation Systems Division Policy and Standards for "On-Line Software Configuration Management"

have been developed and utilized on a number of projects with excellent results.

Computer Printout Examples.

A typical example of one of our programs is shown in the following illustrations. Figure 12 summarizes the DR statistics by system. Figure 13 depicts the last page of the master software DR log. Figure 14 summarizes open DRs by responsible engineer. This printout is utilized at weekly meetings as a basis for review status. Figure 15 summarizes the DR statistics by responsible engineer.

References

Bersoff, Edward H.; Henderson Vilas D.; and Siegal, Stan G. CTEC Inc.; Software Configuration Management: A Tutorial

MIL-STD 1644 (TD), 7 March 1979,
Trainer System Software Development

T44 DR STATISTICS				
SYSTEM	OPEN	TESTED	CLOSED	TOTAL
ENGINE START	0	0	11	11
GROUND DYNAMICS	0	0	17	17
ENGINE GROUND	2	0	2	4
ACCEL, TO, STOP & DECEL	0	0	1	1
FLIGHT INSTRUMENTS	1	0	9	10
WHEELS & BRAKES/CRASH	0	0	4	4
PROPS	0	0	4	4
ATMOSPHERICS/ROUGH AIR	0	0	2	2
ENGINE FLIGHT	0	0	13	13
STALL AND BUFFET	0	0	1	1
ELECTRICAL SYSTEM	2	0	34	36
FUEL SYSTEMS	0	0	13	13
GEAR & FLAPS	0	0	16	16
ANNUNCIATOR & FIRE SYSTEM	0	0	0	0
ECS & MISC SYSTEMS	1	1	13	15
HARDWARE	23	0	126	149
DRED	1	0	10	11
EXEC & DEBUG	1	0	6	7
CONTROL LOADING	2	0	20	22
INSTRUCTOR STATION	23	1	80	104
NAVIGATION	4	1	26	31
NAV RECEIVERS	0	0	14	14
MOTION	2	0	10	12
AUTOPILOT	1	0	6	7
FLIGHT & AERO	3	0	23	26
INPUT/OUTPUT	4	0	11	15
COMMUNICATIONS	3	0	2	5
AURAL CUE	0	0	2	2
PITCH ON GROUND	0	0	0	0
SURFACE VS CONTROL DEFLT	0	0	0	0
PROPELLER THRUST	0	0	0	0
CONTROL-FRICTION/INERTIA	0	0	0	0
TRIM	2	0	0	2
LONGITUDINAL STABILITY	0	0	0	0
LATERAL STABILITY	0	0	0	0
SPIRAL STABILITY	0	0	0	0
REPLAY	0	0	1	1
NCS-31 SELF TEST	0	0	0	0
ATP VERIFICATION	0	0	0	0
INSTRUMENTS	9	0	5	14
TOTALS	84	3	482	569

Figure 12. T-44 DR Statistics for Each System

MASTER SOFTWARE DR LOG						
DR NUMBER	SYSTEMAFFECTED.....	RESP ENG	DATE .TESTED.	DATE .CLOSED.	DESCRIPTION	STATUS TAPE NO.
561	TRIM	DDD				
562	HARDWARE	JR	12/12/79	12/14/79	ELEC TRIM - A250 BOMBS - #2	CLOSED
563	HARDWARE	JR			#2 : SYS FRZ - FUEL FLOW GLITCH	
564	HARDWARE	JR	12/12/79	12/14/79	#1 : VEL PARM FRZ LITE OUT	
565	COMMUNICATIONS	TJD			#2 : WPT REMOTE UNIT SW IS BENT&LOOSE	CLOSED
					LOWEST FREQ IS 225.00	
566	ELECTRICAL SYSTEM	TJD			DC PWR. - GS, PP, CP	
567	INSTRUCTOR STATION	RA			GCA STATUS DISPLAY	
568	INSTRUCTOR STATION	RA			XCY SPECIAL SYMBOLS	
569	HARDWARE	JR			#2 : RUDDER PEDALS	

Figure 13. Master Software DR Log

RTM DR LOG						
DR NUMBER	SYSTEMAFFECTED.....	RESP ENG	DATE .TESTED.	DATE .CLOSED.	DESCRIPTION	STATUS TAPE NO.
426	FLIGHT INSTRUMENTS	RTM				
441	INPUT/OUTPUT	RTM			MAX ALLOWABLE AIRSPEED @165 IS 172	
465	NAVIGATION	RTM			CPLT VSI OFF UP TO 50 FPM	
466	NAVIGATION	RTM			IC5/APPR = IEFD	
478	EXEC & DEBUG	RTM			IC5/ICRP = X-CNTRY NOT APPR	
					MUST CLEAN UP DISK	
556	DRED	RTM			DRED TEST #7 NOT WORKING	
559	INPUT/OUTPUT	RTM			RESET NOT DRIVING FD ATT/ADA	

Figure 14. Open DR Log Keyed to Responsible Engineer

T44 DR STATISTICS				
ENGINEER	OPEN	TESTED	CLOSED	TOTAL
RTM	7	0	36	43
JR	26	0	135	161
TJD	9	1	96	106
SD	3	0	41	44
DDD	5	0	19	24
TZ	4	1	35	40
HD	3	0	5	8
GL	2	0	64	66
RA	20	1	51	72
JM	5	0	0	5
TOTALS	84	3	482	569

Figure 15. T-44 DR Statistics by Responsible Engineer

ABOUT THE AUTHORS

Wynn L. Gaylor is Vice President, Systems Engineering with the Simulation Systems Division of Gould Inc. His primary responsibilities are simulated system analysis, system simulation (math models), and simulation engineering (simulation hardware). Prior to this, he was Chief Computer/Simulation Engineering and Programming. He holds a B.A. with a major in Mathematics and an M.S. in Applied Mathematics from Adelphi University and has an additional 30 credit hours in the Ph.D.

Science Program at Polytechnic Institute of New York.

Ronald Murphy is Manager of Software Engineering with the Simulation Systems Division of Gould Inc. He is responsible for all simulator-related software from design to configuration management. Mr. Murphy holds a B.S. degree in Mathematics from Iona College and an M.S. degree in Management Engineering from Long Island University.