[illegible]

C66-00908

[redacted]

3730-6003-RC000
Total Pages: 92

CLASSIFICATION CHANGE

To UNCLASSIFIED
By authority of GDS - EO 11652 Cat 4
Changed by Dolores L. Merritt Date 12/31/72
Classified Document Master Control Station, NASA
Scientific and Technical Information Facility

APOLLO FLIGHT SOFTWARE VALIDATION PLAN

Final Report, Task 4, ASSAP, Contract NAS9-2938 (U)

3730-6003-RC000

9 June 1965

Prepared for

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

Coordinated by:
R. Braslau, Section Head
Space Guidance Analysis Section

Approved by:
for M. Fox, Project Manager

Approved by:
for R.K. Whitford, Director
Inertial Guidance and Control
Laboratory

Approved by:
H. Leon
Assistant Program Manager
ASSA Program Office

[redacted]

TRW SPACE TECHNOLOGY LABORATORIES
THOMPSON RAMO WOOLDRIDGE INC
ONE SPACE PARK · REDONDO BEACH CALIFORNIA

[redacted]

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

Total Pages: 92

ILLUSTRATIONS

1. SUMMARY

This report describes a general Apollo Flight Software Validation Plan which STL recommends for adoption by MSC. A series of flow diagrams describing the detailed steps to be followed for validating the flight software from the mission planning phase through the post flight analysis phase is presented for a representative operational Apollo mission. The software validation requirements are correlated with the planned Apollo mission schedule. Solutions to the validation problem presented by simultaneous primary missions and contingency flight planning are discussed. The recommended documents and simulation programs associated with flight software preparation, testing, control and approval are described. Essential features of selected software testing techniques and the targeting process are presented as an aid to MSC in the validation process.

When the flight software validation plan is applied to the overall Apollo program, certain conclusions should be emphasized. They are:

1) Some form of software standardization is essential to the success of the Apollo software development and validation.

2) This standardization can only be achieved with early efforts to define and develop complete Apollo operational flight software.

3) High confidence in flight proven software requires effective software control through frequent and meaningful design reviews.

4) The Apollo program plan requires sufficient software development and validation resources to sustain at least three simultaneous flight prepration programs.

5) Consideration should be given to redefining the role of the large hybrid physical/digital simulation programs with respect to software validation.

6) Independent qualification testing and targeting verification is recommended for each flight.

7) A formal targeting and verification procedure should be developed to confidently support the short turn around time inherent in the Apollo Master Test Plan.

2. INTRODUCTION

The purpose of this report is to present a general Apollo Flight Software Validation Plan which is recommended for application to Apollo mission preparation efforts. This is the third and final report resulting from a three month study by TRW Space Technology Laboratories (STL) prepared for MSC under Task 4, ASSAP, Contract NAS 92938. The first report, Reference 1, described a general software validation philosophy compiled from STL's extensive experience with weapon system and space system software development and targeting. This report was not constrained to reflect the current Apollo program resources, simulation tools or development plans. The second report, Reference 2, recommended a specific software validation plan for implementation in Flight 202 which has already been in preparation for several months. The Flight 202 plan adapted much of the existing Apollo software validation tools and documentation to minimize the impact of the recommended plan on the flight preparation schedule and resources, yet provides a basis for thorough control and review of the flight software. When implemented it would form the first step to a more encompassing procedure to be developed as the missions and software become more extensive. The general plan described in this report builds on the Flight 202 plan and recommends a philosophy of software "module" standardization. Using this plan, software "modules" are evolved from flight to flight, leading to the operational mission, and taking advantage of the previous flight software validation efforts wherever possible.

The object of this plan is to provide MSC with the means to monitor, coordinate and control the Apollo flight software. This is done primarily by a series of software design reviews in which specific software specifications, test plans, test results and description documents are required to help MSC insure that the flight worthiness of the software has been adequately demonstrated. The responsibility for the development and correctness of the flight software lies with the MIT Instrumentation Laboratory (MIT), but the verification efforts are shared between various participating agencies such as MIT, MSC, NAA/S&ID, GAEC and Raytheon. This plan provides for the coordination of these efforts in a practical manner.

This general validation plan covers the initial mission planning efforts which lead to specifications on the flight software, the development, programming, and testing of the flight equations and logic, the targeting of the flight constants, the fabrication of the memories for the Apollo Guidance Computer (AGC) and the LEM Guidance Computer (LGC), the participation of the software in the ground checkout process, the in-flight verification function and the post flight analysis effort. Provision is made in the plan for contingency flight plans and software change procedures. This plan is limited to the contents of the flight ropes, the flight portion of the hybrid ropes of the AGC and LGC, and the temporary memory in which portions of the flight program are stored. This limitation has been necessary because of the limited scope of this task and the sparcity of the available documentation on these subjects. Similar (but not necessarily identical) validation efforts would apply to the test ropes, the LEM abort computer memory, the real time ground program, and the Apollo support equipment (ACE) computer programs.

It is assumed that the overall mission planning described in Reference 3 is the current best estimate of the Apollo flight test schedule. It is also assumed that sufficient resources will be available among the participating agencies to not only prepare and validate the software on a flight by flight basis, but also to simultaneously anticipate the needs of later missions so as to permit early standardization of software subprograms wherever possible.

This plan is described in six main sections. The considerations of time phasing the software preparation efforts with Apollo flight schedules is described first. Then a typical complete validation procedure is described in the form of flow diagrams which should be performed for every flight but can be accomplished in part during preceding flights. A list of the primary documentation recommended for the adequate development, coordination, testing and control of the software is provided. This is often looked upon as unnecessary work, but has been shown by experience to be an essential part of any software validation effort. A brief summary of the basic simulations and programs recommended for the software preparation and testing effort is included. Some of the programs would not normally be required, but because they exist, they can be of some value in increasing the confidence in the flight software. Selected software test techniques, in particular the equation test, the program checkout and the qualification test, are discussed to amplify the purpose and contents of these tests. Finally the targeting and targeting verification effort is discussed. This process takes on special significance when relatively large differences exist between frequently occurring flights requiring a rapid response time.

The only part of this report which is classified Confidential is the flight schedule references to calendar dates. With the removal of pages 3-2 and 3-4 this report would become unclassified.

3. TIME PHASING OF SOFTWARE PREPARATION

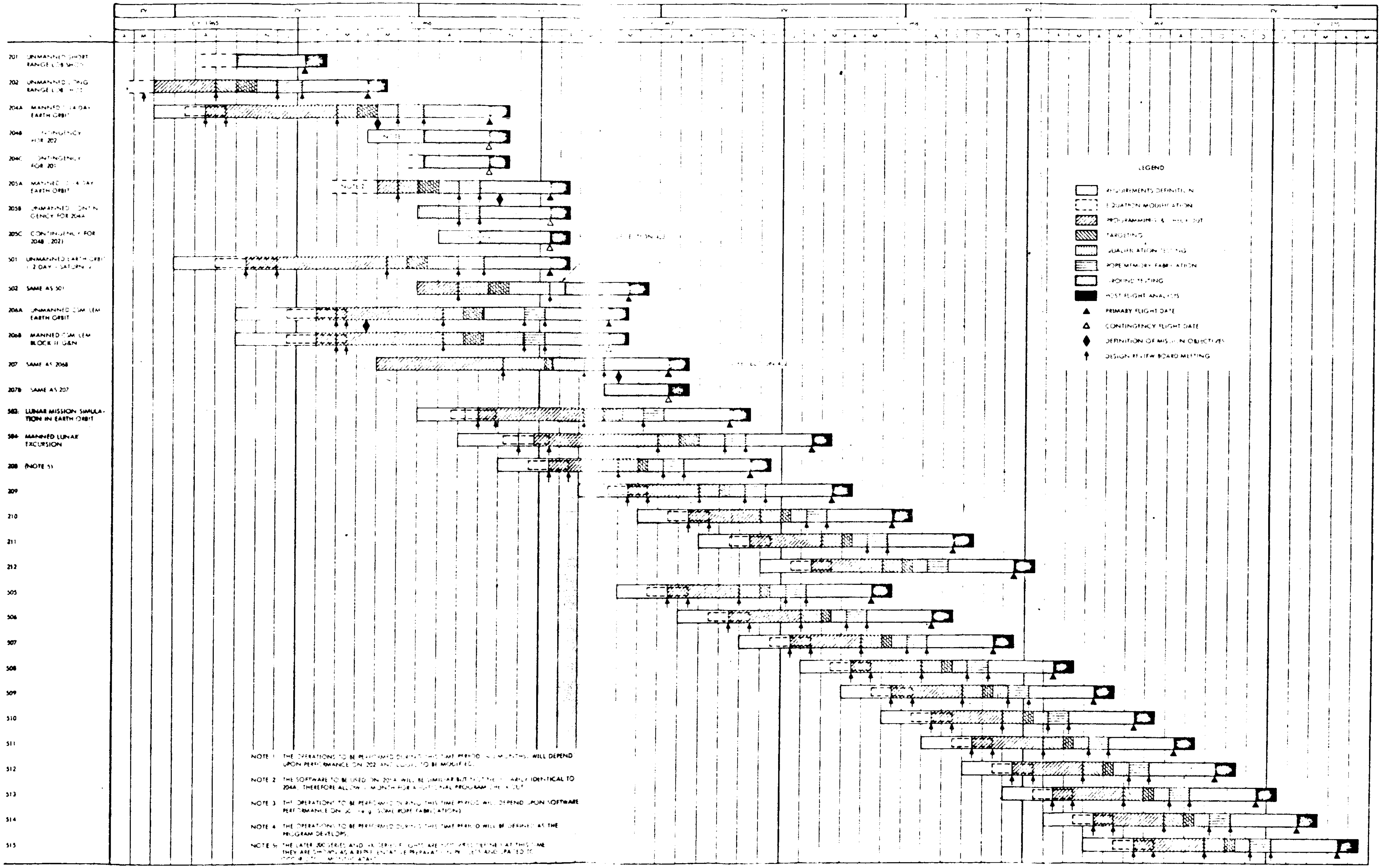
This section contains the time phasing schedule associated with the preparation of software for future Apollo flights. The simultaneous development of software for several flights at one time is a problem confronted early in the Apollo Flight Development Program. Such efforts as preparing or recoding AGC/LGC programs, equations and simulations should be coordinated and the results of each flight should be used in preparing for a subsequent one. The time phasing schedule is designed to permit this coordination. The schedule is intended to show the time period required for software preparation on each Apollo mission and the software preparation problems indicated by contingency and simultaneous missions. The time period of software preparation for each mission was chosen such that the tasks as described in Section 4 can be carried out in an efficient manner.

3.1 FLIGHT PROGRAM DEVELOPMENT SCHEDULE

The overall software preparation and validation schedule for the Apollo Flight Development Program is shown in Figure 3-1.

The flight plan taken from the Apollo Spacecraft Master Test Plan, Reference 3, was used as the basis for this schedule. The schedule is composed of two parts. The first is a nominal program and is defined as the most optimistic program considered feasible with the minimum number of flights. The alternate missions necessary to plan for the possibility of failures in the primary missions, repeated flights and other contingencies are also shown. The time zones assigned to each phase of the software preparation effort were derived from the representative software development schedule as described at the end of this section. The second part indicates the time periods for the undefined 200 and 500 series flights, beginning with missions 208 and 505 respectively. A standard period of 13 1/2 months is shown for each flight because of the lack of information defining the mission objectives. Therefore, this part of the schedule is not realistic since the contingency relationships between each of these flights and those of the nominal program are not available at this time. The real value of this schedule is to indicate the extent of the resources required.

[redacted]



NOTE 1 THE OPERATIONS TO BE PERFORMED DURING THIS TIME PERIOD (N MONTHS) WILL DEPEND UPON PERFORMANCE ON 202 AND [illegible] TO BE MODIFIED.

NOTE 2 THE SOFTWARE TO BE USED ON 205A WILL BE SIMILAR BUT NOT NECESSARILY IDENTICAL TO 204A. THEREFORE ALLOW 1 MONTH FOR ADDITIONAL PROGRAM CHECKOUT.

NOTE 3 THE OPERATIONS TO BE PERFORMED DURING THIS TIME PERIOD WILL DEPEND UPON SOFTWARE PERFORMANCE ON 50[illegible] (e.g. SOME ROPE FABRICATION)

NOTE 4 THE OPERATIONS TO BE PERFORMED DURING THIS TIME PERIOD WILL BE DEFINED AS THE PROGRAM DEVELOPS.

NOTE 5 THE LATER 200 SERIES AND 500 SERIES FLIGHTS ARE NOT WELL DEFINED AT THIS TIME. THEY ARE SHOWN AS A REPRESENTATIVE PREPARATION PROJECTS AND [illegible] TO OCCUR [illegible] MONTHS APART.

Figure 3-1. Overall Software Preparation and Validation Schedule

[redacted]

preparation phase when all 200 and 500 series missions are flown. This is shown in Figure 3-1

From Figure 3-1, the total number of flights involved at the same time for each phase of software preparation is plotted in Figure 3-2. The dark shaded blocks indicate the loading expected when only the defined flights are considered. The light colored areas are the additional effort required because of the undefined missions. Each phase in the nominal preparation program has a peak. As the software effort progresses, the peak shifts to the right. The peaking is the result of contingency and repeated flights occurring within a few months of each other. If a decision is made to accelerate the program and to include additional flights the light colored areas will shift to the left, raising the peak values and possibly expanding the peak to cover a longer time period. The implications on the resources required for each phase will be discussed briefly.

3.1.1 Requirements Definition Phase

The effort required for this phase can be minimized by the standardization of software requirements. The gap in early 1966 is caused by the separation between the planning stages for missions 206, 503 and 504. It can be utilized by starting effort on flights 503 and 504 earlier than shown on Figure 3-1 especially when the status of the undefined flights are defined.

3.1.2 Equation Modification Phase

The amount of work necessary is a function of the level of flight equation standardization. It will be necessary to keep the fully operational software in mind while preparing for the early flights. A simultaneous effort early in the program will result in both reduced effort later and more confidence in the software. Early definition of the undefined missions will assist in efficient use of the minimum effort time period in early 1966.

3.1.3 Programming and Checkout Phase

The broad peak of four flights in this phase at the same time requires a high level of manpower. Scheduling will be a function of the resources at MIT and the standardization obtained with the AGC/LGC flight programs. It is important to note that the load on the computer

[redacted]

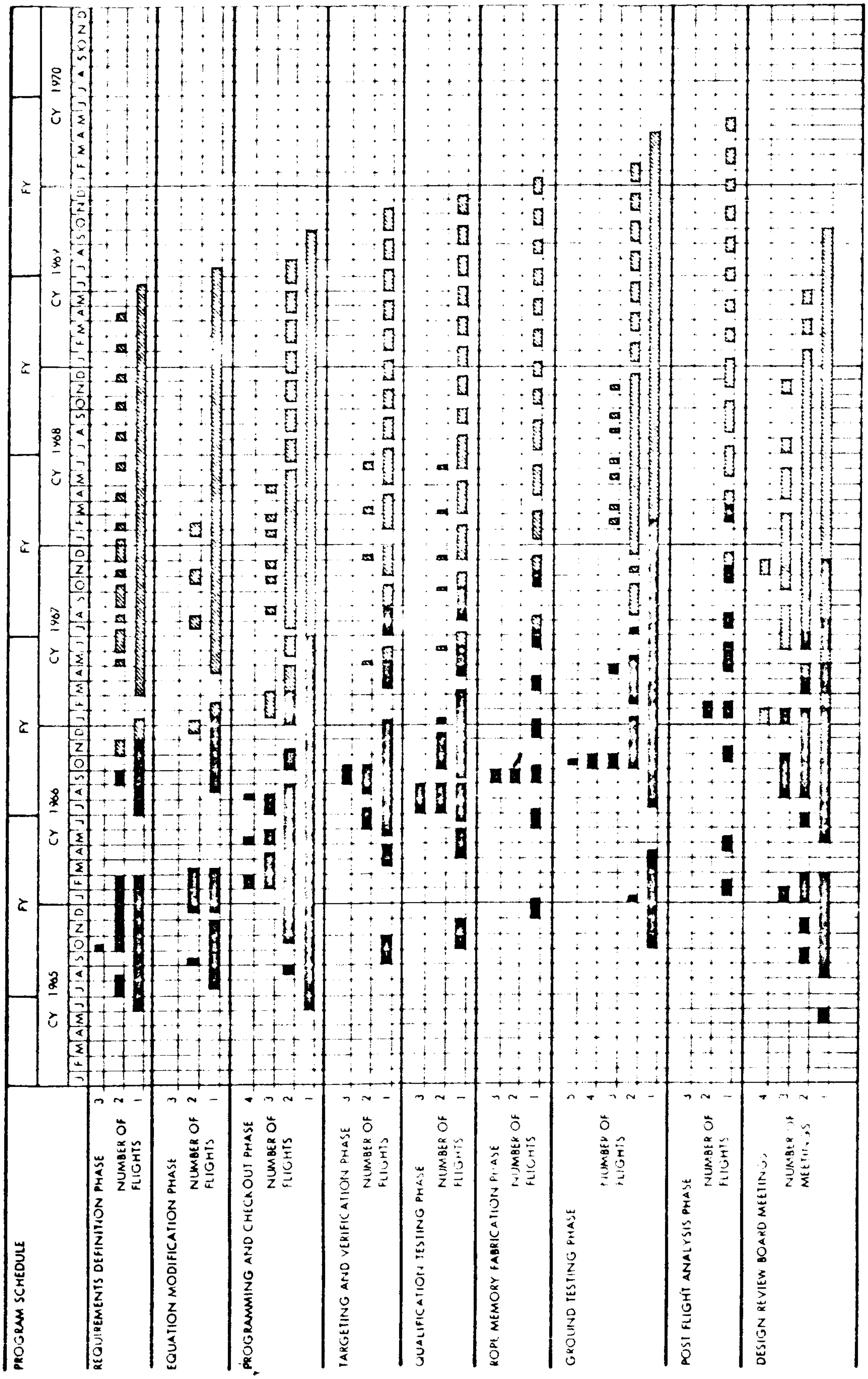


Figure 3-2. Summary of Mission Preparation Phase

[redacted]

facilities will be the sum of the use load required for both the programming and equation modification phases.

3.1.4 Targeting and Verification Phase

The occurrence of simultaneous targeting of those flights at the same time will necessitate some type of semi-automatic program to accomplish the targeting. The frequency of the targeting effort does not lend itself to trial and error targeting methods.

3.1.5 Qualification Testing Phase

Full time use of computer facilities will be required during the peak months. Computer time can be expected to run into actual days of usage in particular if a real time lunar mission simulation is planned.

3.1.6 Rope Memory Fabrication Phase

Ropes for three flights will be in fabrication at the same time. Raytheon manufacturing should plan for this by providing adequate equipment for simultaneous wiring of computer ropes.

3.1.7 Ground Testing Phase

As many as five flights could be in the ground test phase during the month of October 1966. This places requirements on the amount of assembly, integration and checkout facilities required.

3.1.8 Design Review Board

The MSC Software Design Review Board is the principal review body in the software validation process. It should consist of approximately six members and a chairmen. The personnel on the board should be knowledgeable in some aspect of guidance software and not directly related to the software development.

In the nominal flight test program it is possible to have ten different validation processes occurring at the same time with three meetings a month for several months. If the post 207 and 504 flights are included this number can become at least four per month. This would establish the board membership as a full time job. Since the board member will be a responsible individual with other administrative functions, full time effort on design review is not desirable.

It is recommended that two personnel pools be established. The first will be a board chairmanship pool consisting of three to four people. Approximately eighteen other individuals will be assigned to the second or associate member pool. The board meetings will be staffed by people from these pools who will be on duty for a certain period of time. The period will depend on the flight being reviewed at the time. Membership can rotate between flights having major differences in objectives which affect the software. Related flights such as 501 and 502 should have a common board to maintain continuity. It is essential that all documentation and board minutes be distributed to the remaining pool members who are acting as alternates when not attending a software validation meeting. Chairmen can act as an associate member but the reverse case should not occur. This is because the board chairman is usually a more senior individual who will be responsible to MSC to ensure that the flight software has performed its functions on the Apollo missions.

3.2 RELATIONSHIP OF FLIGHT TEST PLANS TO SOFTWARE

Figure 3-3 is a diagram indicating the expansion of software capability from flight to flight. Only the defined program flights are considered. As the Apollo program progresses, additional requirements are placed on the software, but much of the capability necessary for earlier flights can be used continuously. The software for Flight 202 should be systematically expanded and developed to support a fully operational, i.e., a lunar landing mission. Figure 3-3 lists the flight equation and AGC/LGC program subroutines deduced from available documentation. This terminology may change as the routines are modified or combined into sub-program assemblies. It is of importance to note that by use of standard programs and subroutines the effort of developing software for each flight can be minimized. For example, axis transformation subroutines of the In-Flight Alignment program require little modification after being formalized. It can be seen that flights 204A, 206B, 501, 503 and 504 represent a substantial change in mission objectives and therefore require a major updating of the software.

The effort to initialize Block II equations and programming is shown starting with Flight 206B assuming that a Block II CSM/LEM

Flight Equations and AGC/LGC Programs	202	204A	205	206A	206B	207	501	502	503	504
Prelaunch Alignment	●	S	S	S	●	S	M	S	S	S
Booster Monitor	●	S	S	S	●	S	M	S	S	S
Coast	●	M	S	S	●	S	M	S	S	S
Pre-Thrusting	●	M	S	S	●	S	S	S	M	S
Thrusting	●	M	S	S	●	S	S	S	M	S
Mission Control (Uplink)	●	M	S	M	●	S	M	S	M	M
Computer Monitor	●	M	S	S	●	S	M	S	M	M
Abort Modes	●	M	S	S	●	S	M	S	M	S
Mission Sequencer Interface	●	M	S	M	●	S	M	S	M	S
Rendezvous		o	S	S	●	S	S	S	M	M
Thrust Vector Control	●	M	S	M	●	S	M	S	M	S
Attitude Control	●	M	S	S	●	S	M	S	M	S
LEM Landing					●	S	S	S	M	S
Midcourse Guidance					●	S	S	S	M	S
Entry	●	M	S	S	●	S	M	S	M	S
Orbital and Midcourse Navigation	●	M	S	S	●	S	M	S	M	S
InFlight Alignment		●	S	S	●	S	S	S	S	S

Key: ● Initialized
S Same program as previous flight
M Modify previous flight program

Figure 3-3. Flight Equation and AGC/LGC Program Development is available. If this is not the case, such effort will begin with Flight 207.

In general earth orbital missions will not require launch time dependent constants and would not require retargeting unless the mission profile changes, the vehicle and equipment models change, or the software changes as indicated by previous post flight results. Retargeting will always be necessary for lunar missions because of the changing earth-moon geometry, even if the mission is identical to a previous one in every other way. In these cases, only the targeting verification need be performed and the rope module containing the new constants fabricated and checked out.

Each mission will be discussed briefly to identify the flight software implications.

3.2.1 Mission 204A

This is the first manned orbital flight and is further distinguished from flight 202 in that a docking maneuver and significant plane change maneuvers are planned. This requires additions to the Flight 202 software in the Astronaut/DSKY/display area, as well as the rendezvous and docking routines. The extended application of the navigation equations will require additional testing if not modification. The manned mission procedures will also be applied for the first time. However, much of the preflight, boost monitor, orbital thrust control, navigation, navigational update, and re-entry logic developed for Flight 202 can probably be used for this mission.

3.2.2 Mission 205A

This mission will be a long duration manned flight with objectives of 204A repeated. Software for 204A may be adequate for 205A perhaps with frequent ground updating of the navigation computations as indicated in Figure 3-2.

3.2.3 Missions 206A, 206B and 207

A special case exists for Flights 206A, 206B and 207. Because of the substantial difference between Mission 206A and 206B, a parallel effort is recommended up to the indicated decision point as shown on Figure 3-1. At this time it will be necessary to specify the mission objectives for 206. Mission 206A is an unmanned flight to test a complete LEM system. Mission 206B is the earliest possible manned flight of a complete Block II CSM/LEM system. If the Block II CSM is ready or expected to be available, effort on 206A will cease and planning for 206B will continue. However, if this is not the case, data prepared for 206B will be shifted to support flight 207 and preparation for Mission 206A will continue.

Software preparation for 206B requires recoding all equations and revision of the interpreture computer simulations to a Block II configuration. Modifications will be necessary because additional requirements of the CSM/LEM rendezvous and CSM/S-IVB attitude control equations. Planning must also include the basic LEM vehicle simulation and LEM attitude control equations.

3.2.4 Mission 501

This is the first Saturn V flight test in support of the Apollo Flight Development Program. The launch vehicle booster guidance monitoring equations and boost abort logic may require modification. Some modifications may be required in the prelaunch equations. The mission sequencing logic may also be affected by the change to the Saturn V booster configuration. Because of the change of launch vehicles, simulation models, disturbance effects and mission control procedures will be affected.

3.2.5 Mission 502

This mission has the same objectives and mission description as Mission 501.

3.2.6 Mission 503

This is the first manned Saturn V flight test. Its objective is to stimulate the entire lunar mission in earth orbit as much as possible, providing a mission profile that will result in adequate post-flight data. The complete set of lunar mission software must be provided and tested to determine its compability with the earth orbital flight. The mission control (up-link logic) may require modification to optimize the telemetry capability.

3.2.7 Mission 504

This flight has been designated as the first possible opportunity for a lunar landing. Consequently, the software at this point will be at its maximum or fully operational capability to support this mission. Any modifications indicated by the post-flight data from Flight 503 may have to be incorporated in the software supporting Mission 504.

Additional planning and testing for contingency Flights 204B and C, 205B and C and 207 are shown in Figure 3-1. One reason for the relatively short time periods is that the possibility of a substitute flight may not become evident until the post-flight data of the primary mission becomes available. A second reason is that the mission objectives of the contingent flight are either less complex than the primary mission or the software supporting the back up mission would in most cases already be available.

3.3 REPRESENTATIVE SOFTWARE VALIDATION SCHEDULE

This section contains an example of a typical software validation schedule as indicated in Figure 3-4. The schedule and each major step in the schedule will be adjusted for each mission depending on the proportion of new requirements defined, the amount of new sub-programs being developed, and the complexity of the mission.

The software preparation process is periodically reviewed between one and four month intervals. Where the flight program will not change from that developed for a previous mission, the qualification tests would only reflect updated vehicle and equipment model effects, and rope fabrication could possibly precede targeting. However, the usual situation would require some preliminary targeting to precede qualification tests and both processes would then occur roughly simultaneously. The targeting verification is primarily intended to verify the specific flight constants and satisfaction of specific mission requirements, whereas the qualification tests are primarily for the verification of the standardized flight program in all its intended applications.

The schedule calls for an early copy of the flight program deck and target dependent constants to be sent to NAA/S&ID and GAEC for incorporation into their hybrid physical/digital simulation core rope simulators. These hybrid simulations are then used to verify the vehicle and equipment models used during the qualification test. The results are reviewed at the flight program release review meeting. The dashed lines following the hybrid physical simulation tests indicate the further use of these programs with a flight rope for purposes other than flight software validation. These purposes are not understood by STL, but may be used to assist in ground program assembly and checkout.

The flight rope, delivered three and a half months before launch, is then processed through the ground assembly and test, as reviewed by the Flight Readiness Committee. The spare flight rope is checked by MIT using their hybrid simulation before sending it to Florida for storage.

The flight preparation continues through launch, flight operations and post flight analysis.

A detailed software validation milestone schedule similar to that of Reference 2 could be provided here also. However, not all of the details are necessarily prepared specifically for each flight. These details could best be provided during the requirements definition phase for each flight and coordinated by all concerned.

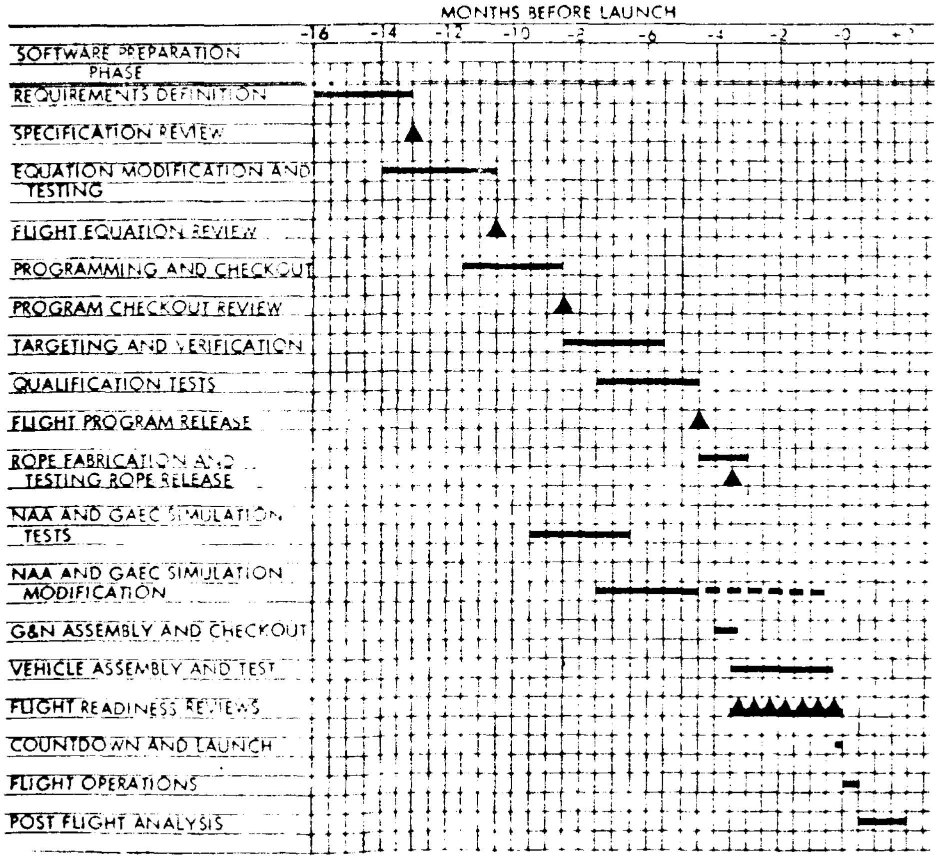


Figure 3-4. Representative Software Development Schedule

4. REPRESENTATIVE COMPLETE VALIDATION PROCEDURE

The complete validation procedure described here is intended to exercise and verify the expected operational performance of the command module and LEM vehicle flight guidance software for all anticipated Apollo missions. The flight software includes the pre-flight and in-flight guidance equations, special systems checkout routines, mission and launch time dependent parameters, IMU sensor flight compensation quantities, flight computer diagnostic checks, and, special routines for testing the G&N operations, interfaces, and hardware performance. In addition to the flight software, the validation procedure is concerned with the targeting program, the rope fabrication and verification, the necessary verification simulations, and validation documentation.

~~tion documentation.~~

4.1 DETAILED VALIDATION STEPS

Since the ultimate purpose in the validation procedure is to obtain the greatest possible confidence in the flight software contained in the AGC and LGC, the procedure has been constructed to emphasize rigorous testing of all areas pertinent to the flight program. In Figure 4-1 the procedure has been divided into nine distinct steps. In general the steps are terminated in a critical design review by the MSC Software Design Review Board. It is the responsibility of this board to ensure that the flight software has satisfactorily met all tests objectives in each step of the procedure before continuing to the next step. Also, the board should perform a complete examination of all test plans and test results documentation.

In Step 1, the overall mission planning as it relates to the Software Requirements Definition is given. The design review here will be responsible for approving the flight software specification and equation test plan. Step 2 consists of the AGC and LGC flight equation development modification, and testing. It ends with a design review to certify that the preliminary flight equations will result in a satisfactory completion of the mission requirements. The programming and checkout of the flight program assemblages are contained in Step 3. The guidance program which results from this step is considered to be the basic flight

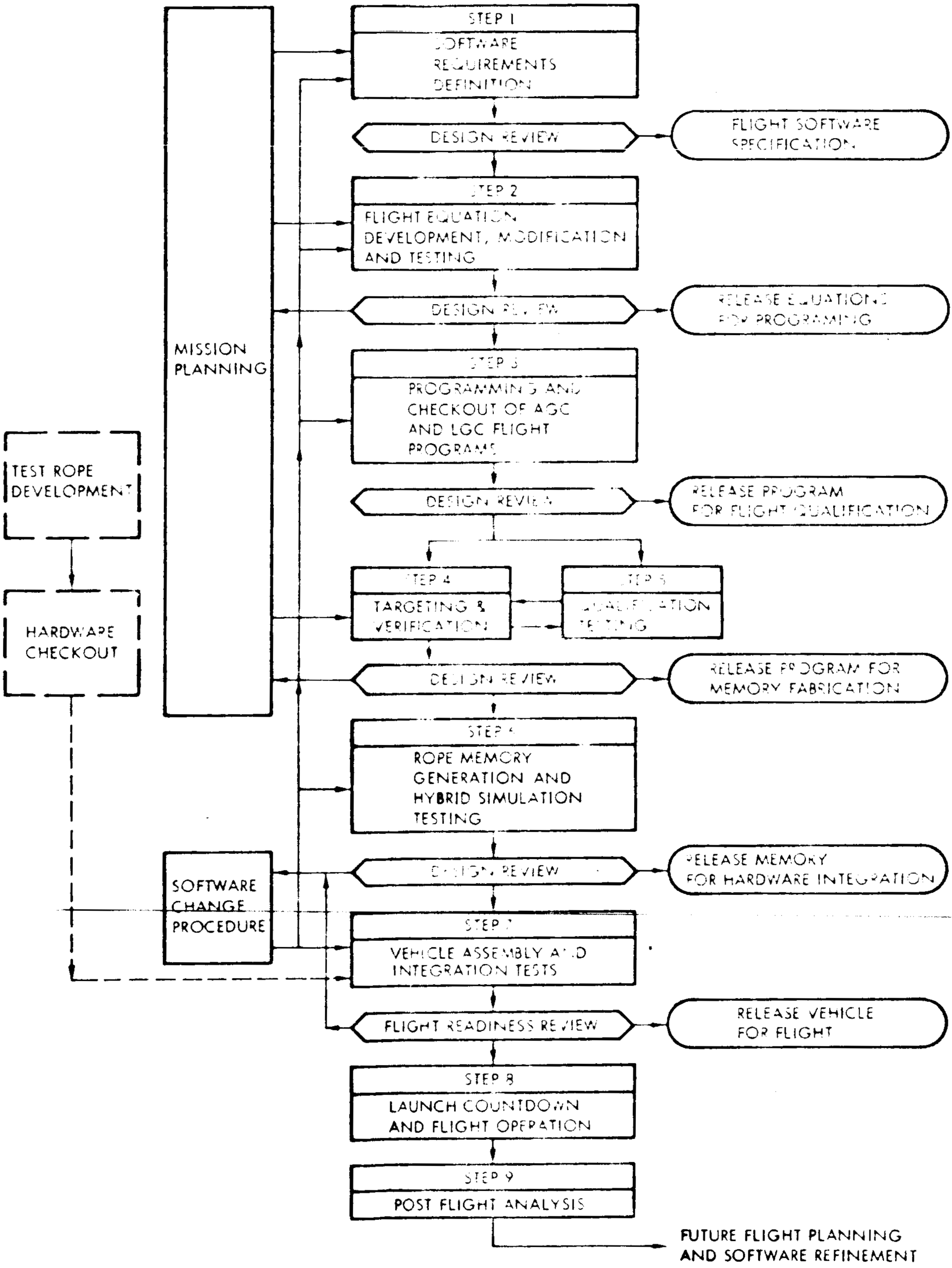


Figure 4-1. Recommended Software Validation Procedure

program for the specified mission. Steps 4 and 5 are devoted to targeting and verification and, program qualification testing. The changes which might result from the targeting and qualification testing processes are expected to be minimum and can be implemented relatively easily with the necessary revalidation of previous steps included in the qualification tests. It is necessary to perform extensive qualification testing only for the completely new software modules. If the flight configuration has not changed, this means that only the destruct readout memory (DRO) must be generated. Therefore, Step 5 can be modified to include only a minimum amount of qualification testing. Upon completion of the mission targeting, the AGC and LGC programs will remain fixed. Only those changes determined necessary by the MSC Software Design Review Board for satisfactory completion of the flight objectives will be permitted. In Step 6, the AGC and LGC memory core rope modules are fabricated along with the generation of the DRO memory punched tapes. The fabricated memories are tested in hybrid simulations to ensure compatibility with the released programs in Step 4. The core rope modules are integrated into the AGC and LGC in Step 7 as part of the G&N system assembly and checkout. System integration tests are conducted on both component and system level of the G&N equipment. The tapes containing the variable memories are loaded and verified by means of the ACE program into the command module and LEM computers. This is followed by an extensive series of ground tests which are performed during vehicle assembly, integration, and checkout. The Flight Readiness Review will notify the launch operations control to begin the countdown and launch procedure in Step 8. The Missions Operation Control Center would monitor the launch operation and take control thereafter. During the flight, software validation continues in the on-board computers and through the telemetry link in the real time ground programs. In Step 9, the post flight analysis is performed. The results of this effort are used in preparing for subsequent flights and also for refining hardware and software design, and performance estimates.

As shown in Figure 4-1, the development of the test ropes, which are used in the hardware checkout, follow a similar validation procedure to that of the flight software. The test rope program should be designed after a definition of test requirements and equipment configuration are defined. Program checkout and rope fabrication verification tests should be performed before the test ropes are incorporated into the guidance hardware for component and subsystem checkout and acceptance. No extensive simulation testing of the test rope programs is required as the ground test itself can provide a realistic test of the program's adequacy.

Figures 4-2 through -10 present the details of each step in the total software validation process for the Apollo missions. Operations to be performed are indicated by rectangular boxes. All documentation is indicated by a rounded figure. A hexagonal figure denotes a review point. Solid lines from the various operations indicate a direct flow process. The dashed lines refer to feedbacks from the review functions.

The software change procedure is an effective way to properly isolate, analyze, and implement the necessary software corrections into the AGS and LGC programs. As indicated in Figure 4-1, the software change procedure can be applied during any of the steps through 7. Any modifications to the mission planning after the software targeting has been completed will be of the form of software change as noted. The flow diagram for the change procedure is presented in Figure 4-11.

4.1.1 Flight Software Requirements Definition

The flight software validation procedure begins with the simultaneous operations of preliminary mission planning and an analysis of the specific software requirements as shown in Figure 4-2. This first phase of mission planning is concerned with defining the preliminary mission requirements, constraints, reference trajectory and schedules. This planning is performed by ASPO at MSC with the coordination of other organizations within MSC, MSFC agencies, and the prime contractors. The preliminary software requirements analysis involves the determination of the required software subprograms which are not currently available in the MIT guidance program library.[†] This effort is based upon developing standarized guidance program modules which can be used throughout the Apollo flight series. Thus, only software

---

[†] This library of subprogram assemblages is called the "Sunrise" and "Corona" series.

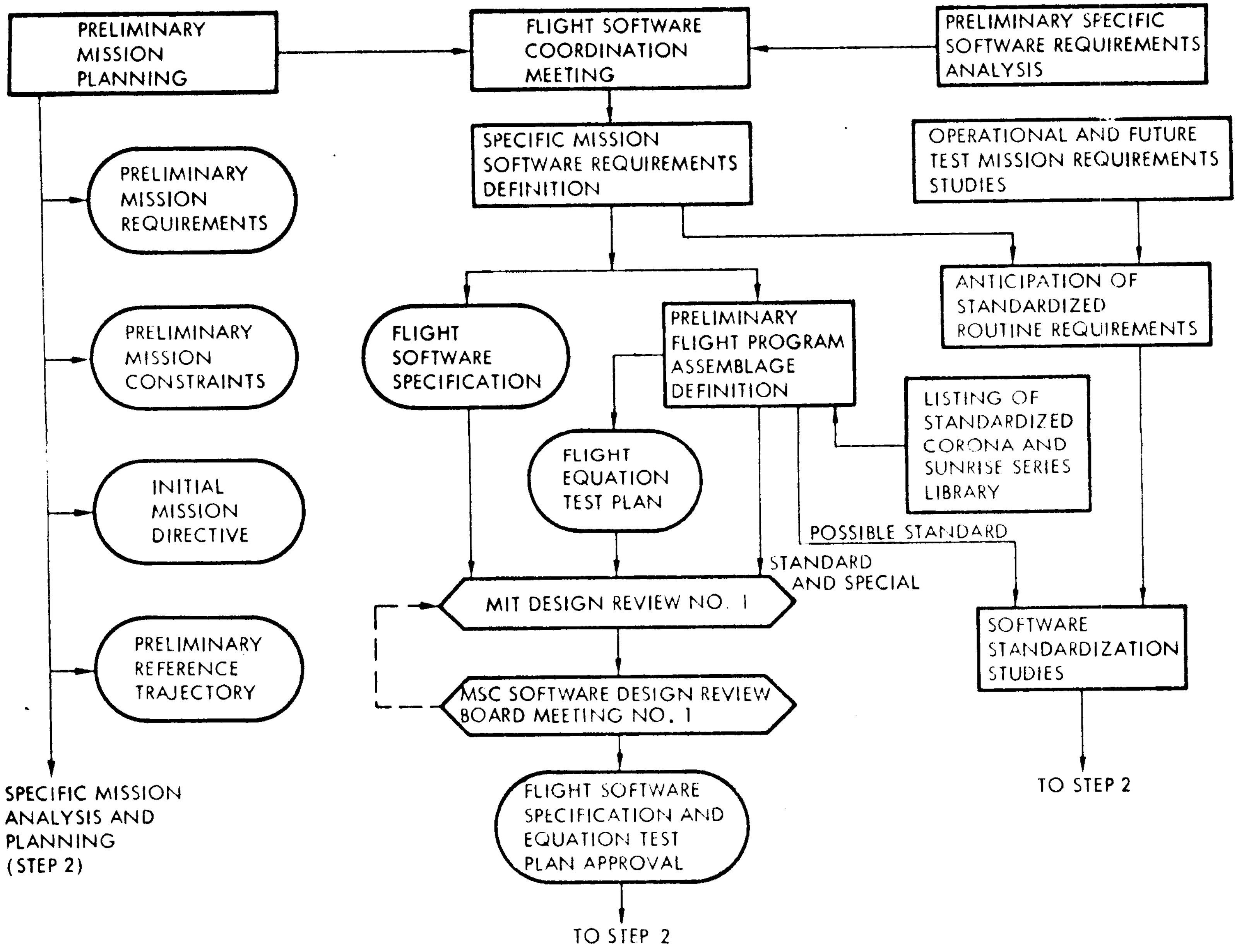


Figure 4-2. Step 1: Flight Software Requirements Definition

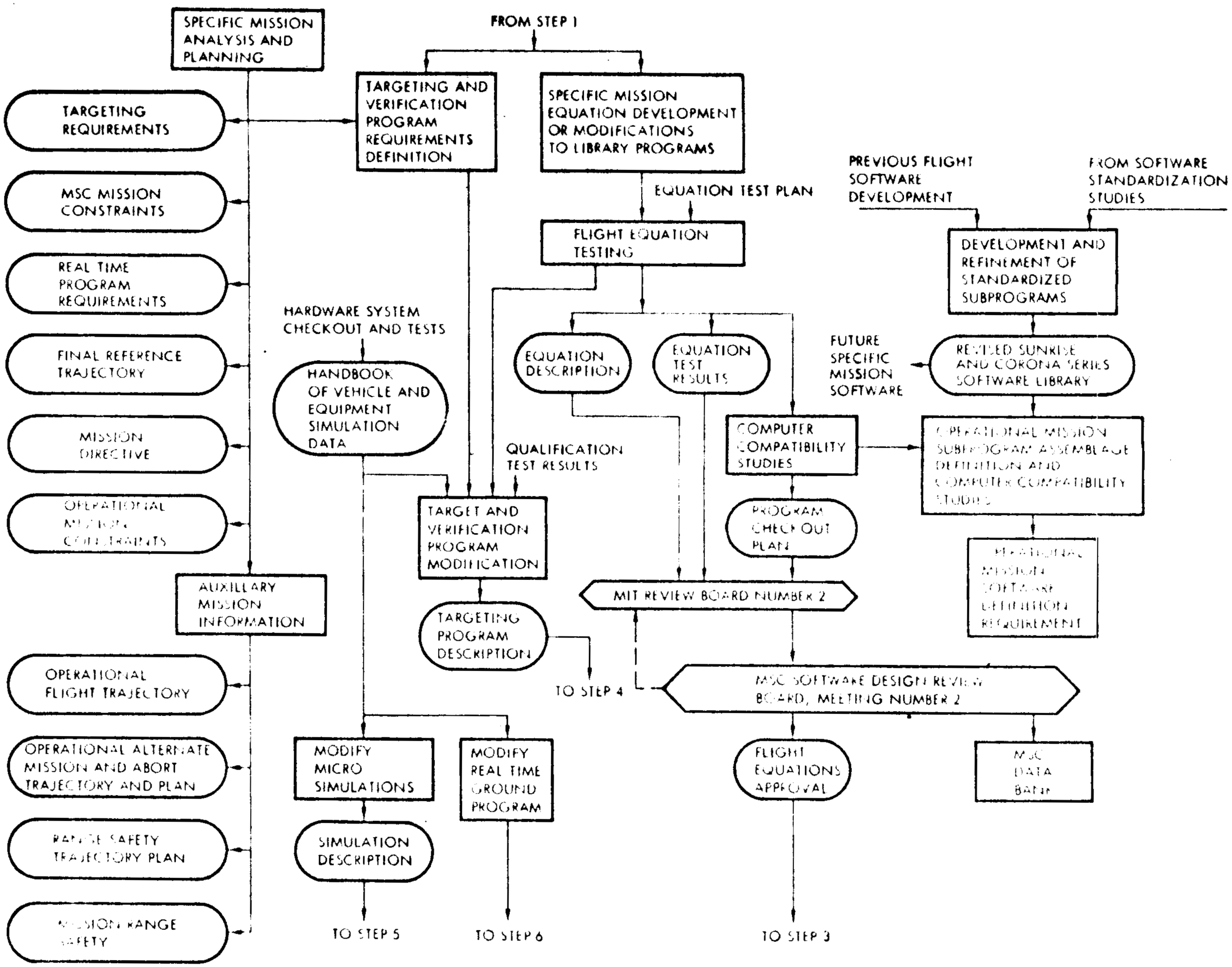


Figure 4-3. Step 2: Flight Equation Modification and Testing

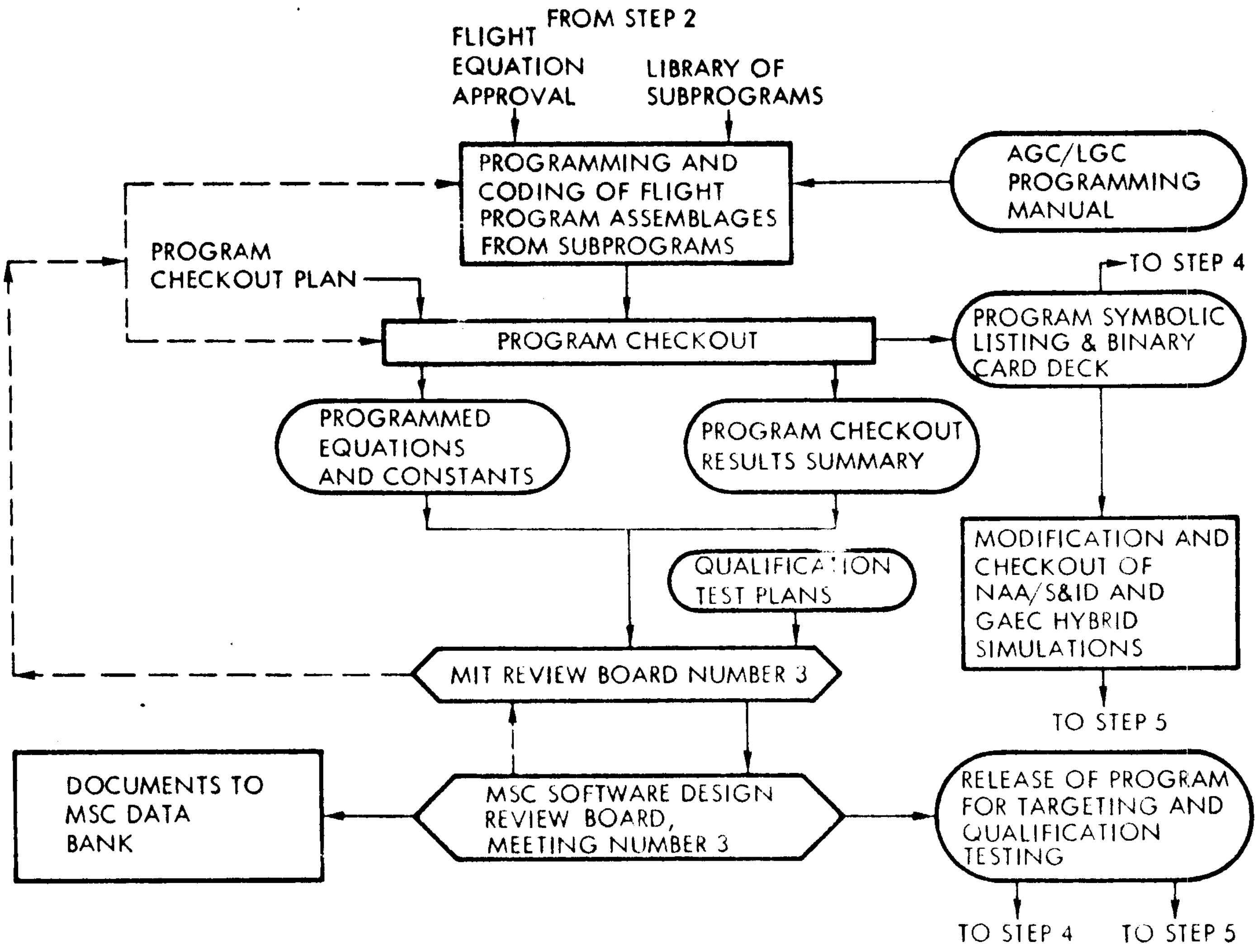


Figure 4-4. Step 3: Programming and Checkout of Flight Equations

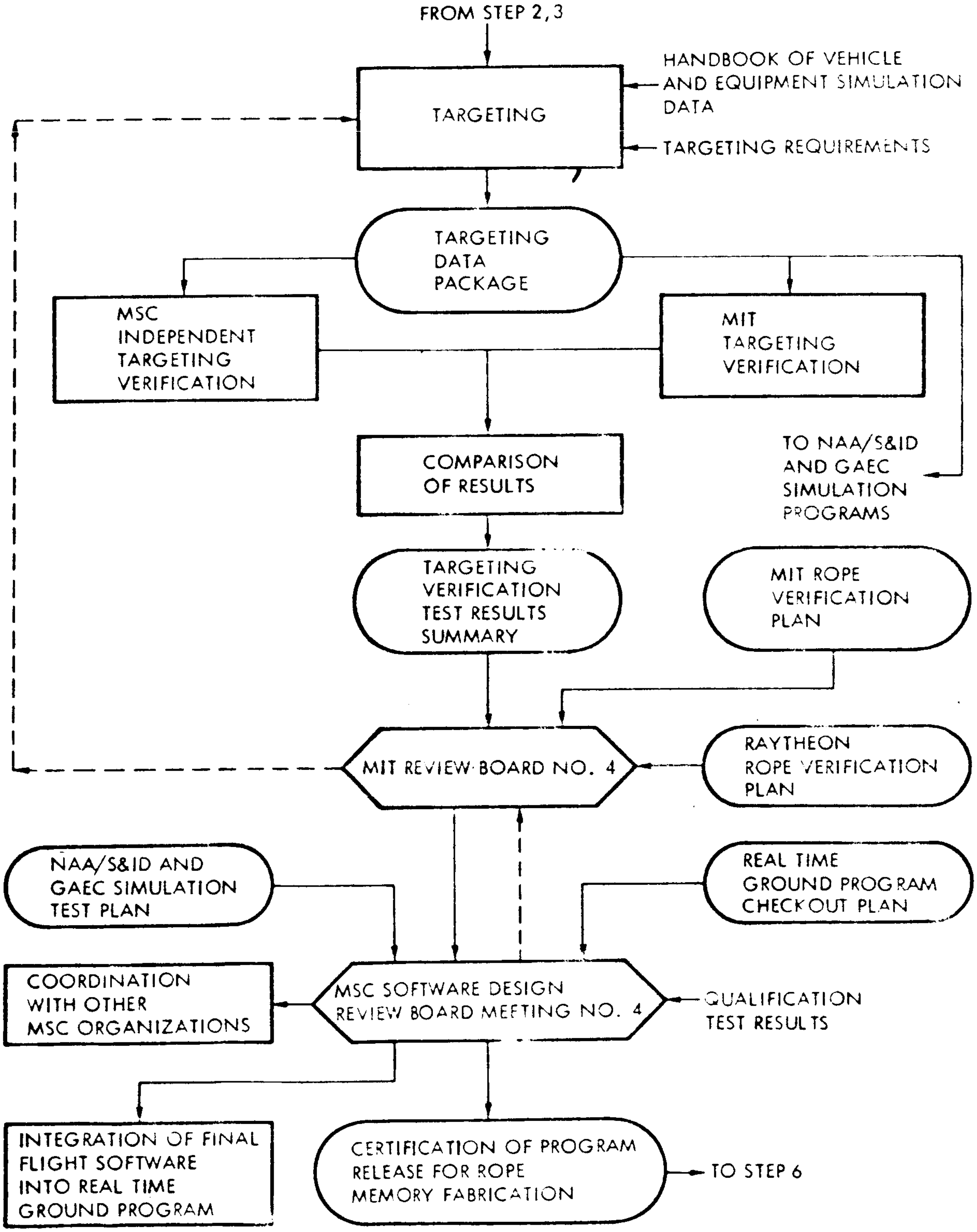


Figure 4-5. Step 4: AGC/LGC Program Targeting and Verification

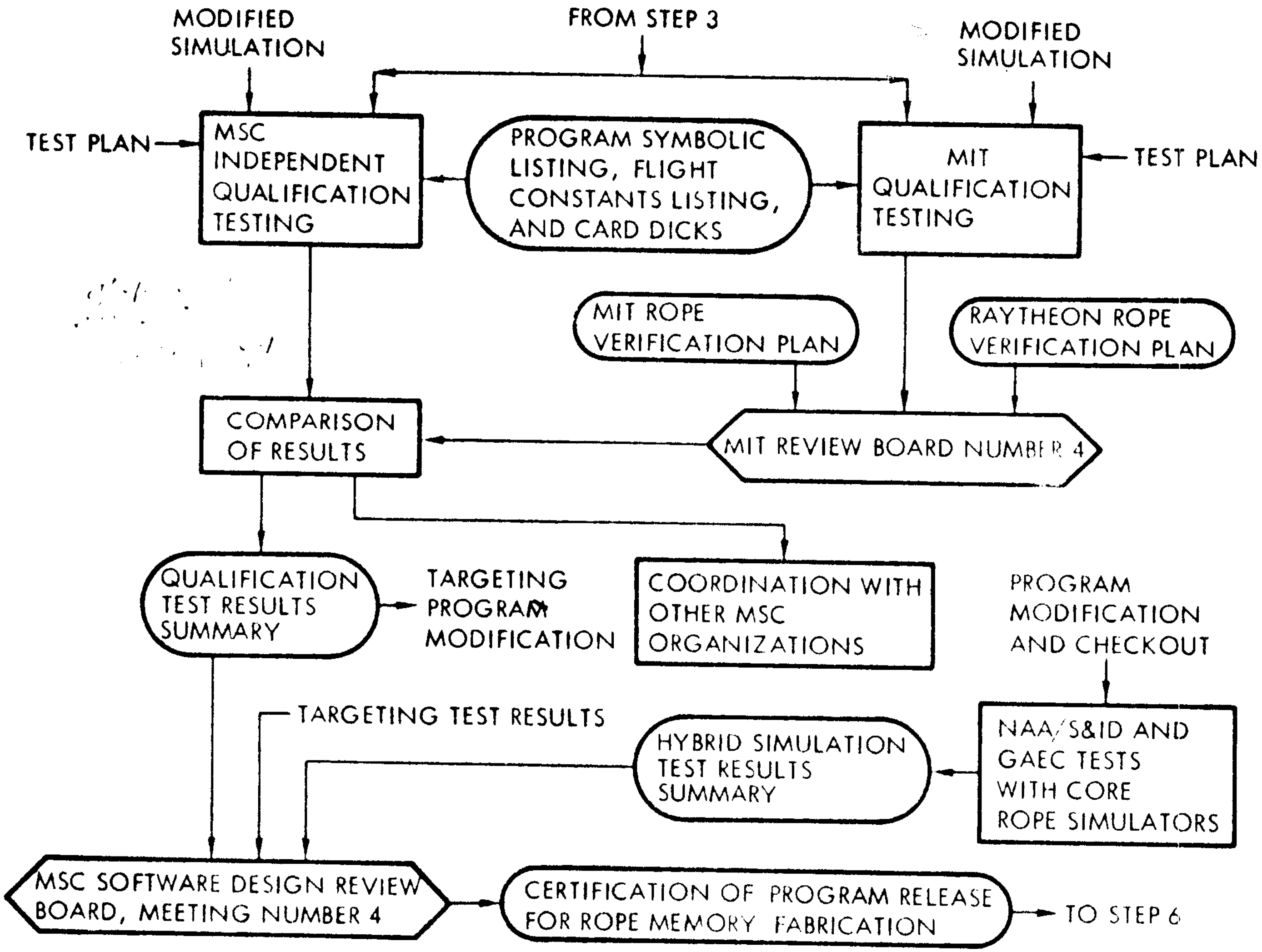


Figure 4-6. Step 5. AGC/LGC Program Qualification Testing

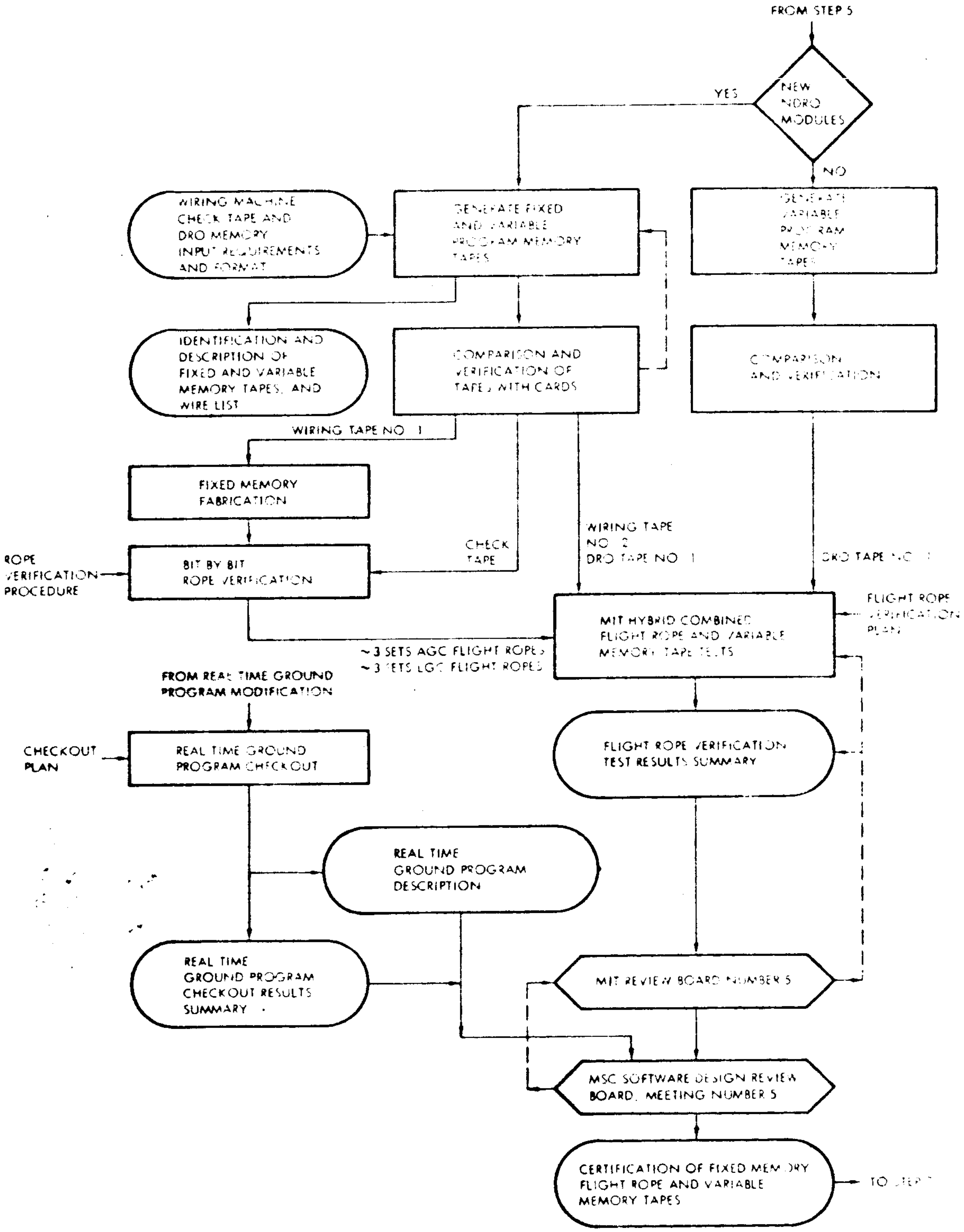


Figure 4-7. Step 6: Rope Memory Generation and Hybrid Simulation Testing

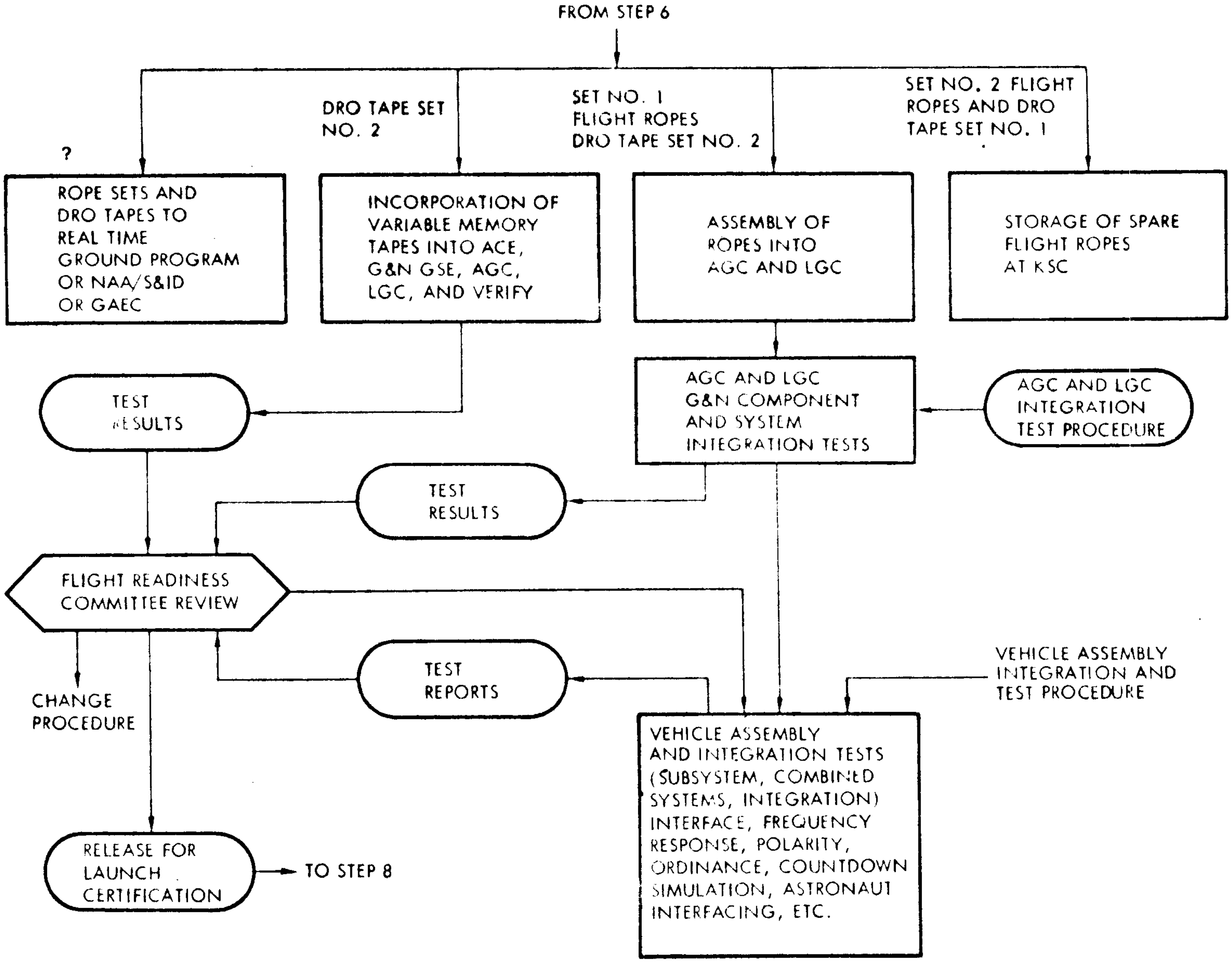


Figure 4-8. Step 7: Hardware Integration and Ground Testing

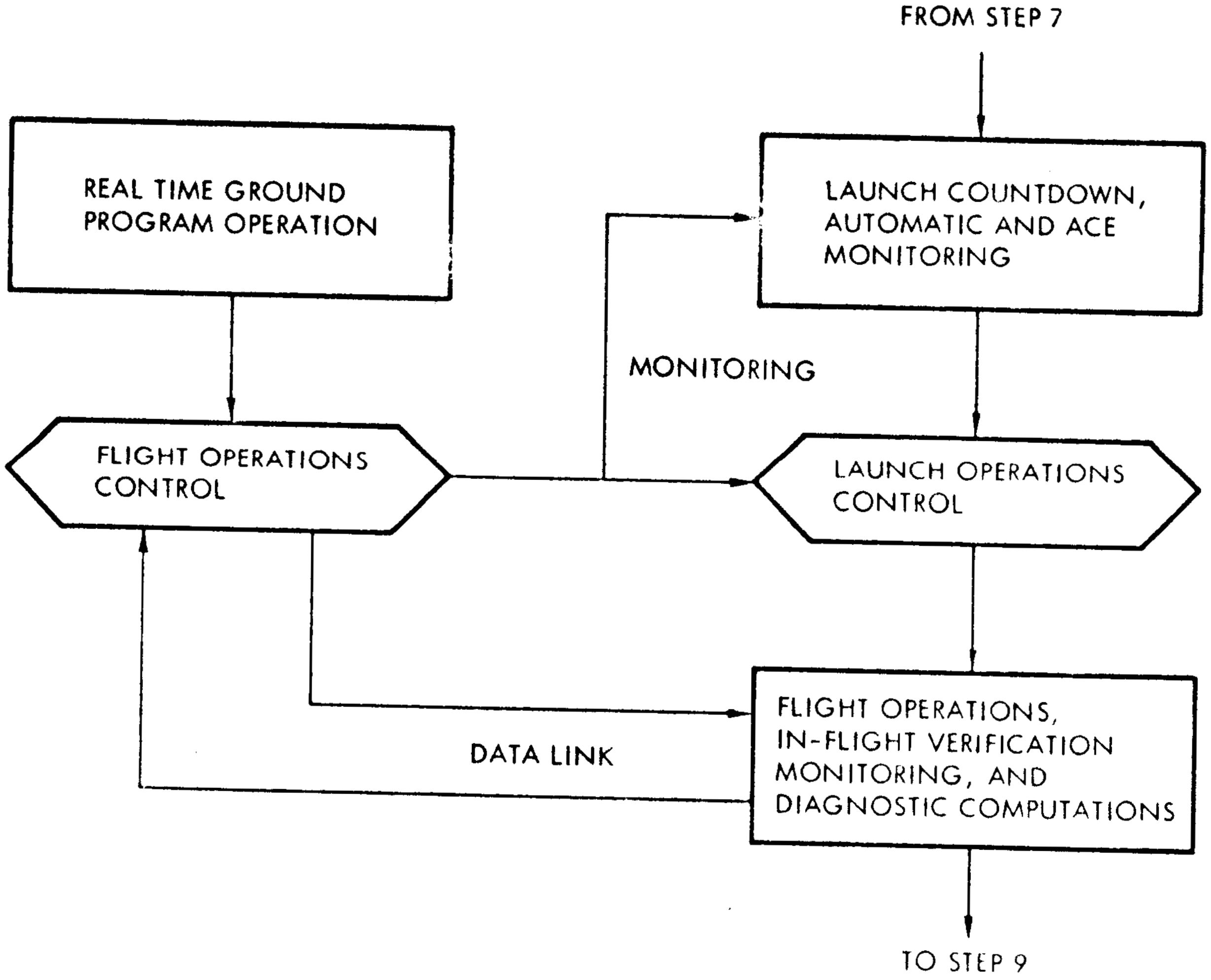


Figure 4-9. Step 8: Launch Countdown and Flight Operations

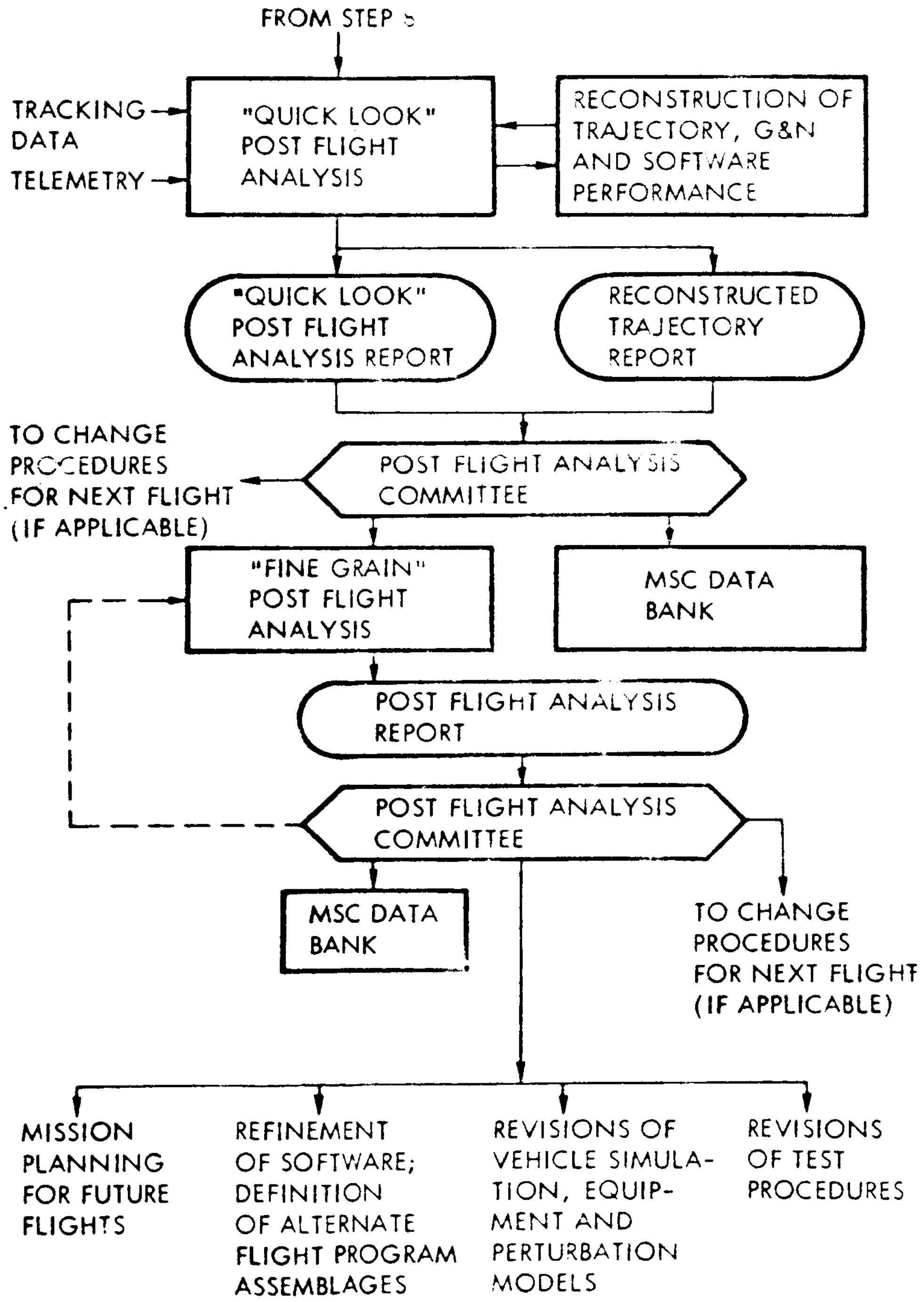


Figure 4-10. Step 9: Post Flight Analysis

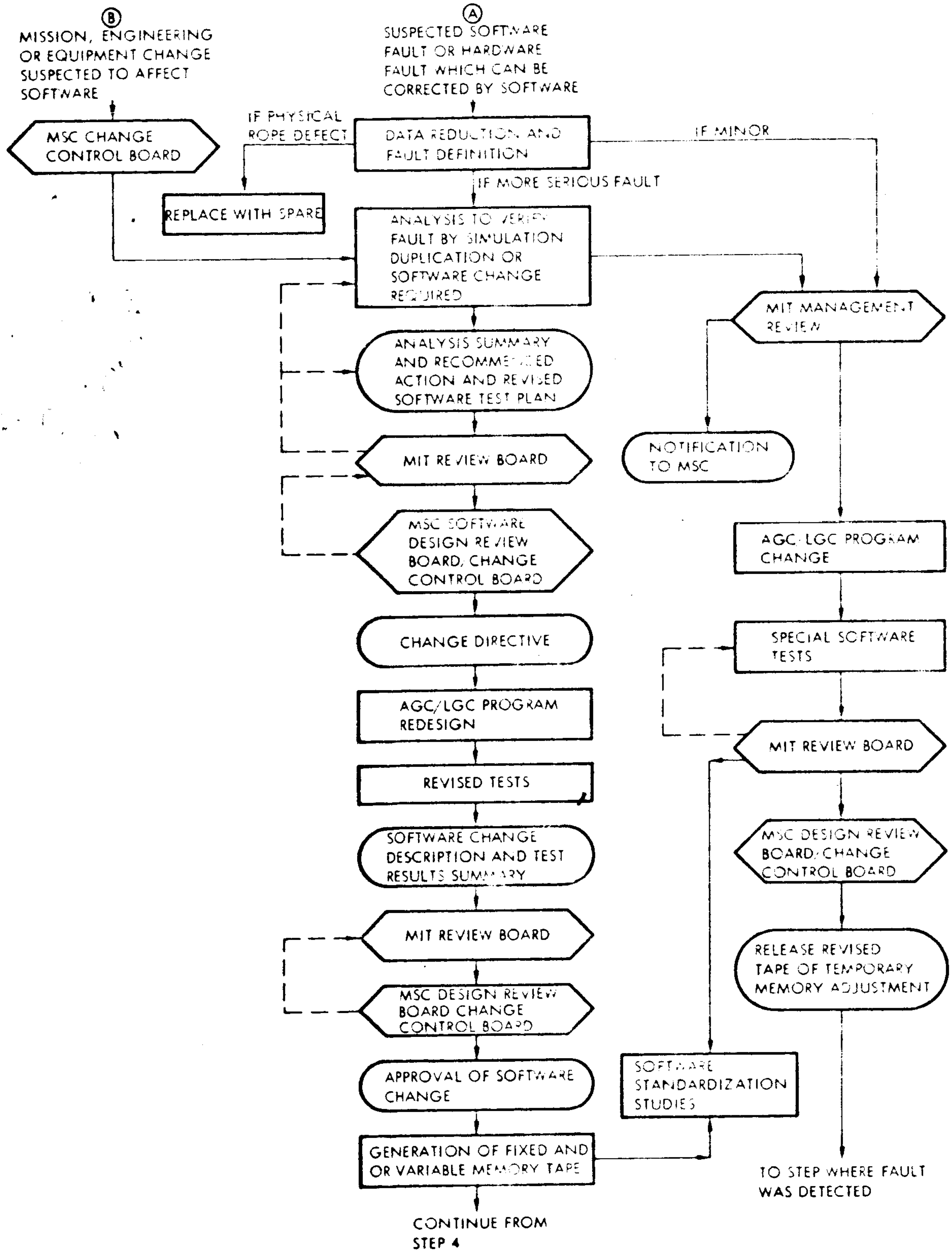


Figure 4-11. Software Change Procedure

program development would be required for specific mission usage or for special one-time use on a particular flight.

The results of these analyses are used to define all of the specific mission software requirements and for generating the Guidance Software Specifications for the command module and the LEM vehicle. These specifications will include software requirements under nominal conditions, performance specifications for non-nominal conditions, determination of backup modes, designation of perturbations for the non-nominal operations, vehicle, IMU and computer simulation models, and specification of the methods and objectives of all test plans required in the validation process. The specification, although specifically applicable to a given flight, should evolve from the previous specification with only the necessary changes.

The output of the above analyses is also used together with some of the standardized guidance subprograms in the Sunrise and Corona library to assemble a preliminary form of the flight program assemblages. From this information, a Flight Equation Test Plan can be generated. This test plan, together with the software specifications and program assemblages is then reviewed by the MIT and MSC Design Review Boards. The MSC board is the principle reviewing panel in the software validation process. The primary responsibility of this board is to ensure that all flight software objectives have been satisfactorily completed as designated in the specifications and test plans. It also provides a software control and coordination function.

The final output in Step 1 is a letter of certification by the MSC board which indicates the satisfactory completion of the developmental step reviewed.

It is recommended for the software validation process that the guidance subprograms used for generating the complete flight programs be standardized. That is, for any operational Apollo mission the required software programs can be assembled from a library of existing subroutines as opposed to developing separate guidance equation programs for each flight. This allows a significant advantage in the program development lead time which is very important for operational mission when flight cycle time may be as frequent as two months. The standardization of the software routines is a continuing effort throughout the validation process. In Step 1, the operations are shown in which future and operational mission software requirements are studied as a separate parallel effort along with the specific flight under preparation. This study effort continues into Step 2 where the standardized subprograms are developed and refined according to current and anticipated program requirements.

4.1.2 Flight Equation Modification and Testing

The second principle phase in mission planning is the specific mission analysis and planning as shown in Step 2. It is during this step that the target requirements, and reference trajectory are finalized. The inputs used in these determinations oritinate from the contractors, within MSC, and from the software development. The primary output documentation is shown in Figure 4-2. This data is used in the continuing software development process where needed. The software targeting program is defined and modified based on the equation required and the outputs of the mission planning. A description is provided including the technical requirements for the program, general operation modes description, simulation flows, major subroutines and interface description, required outputs, and descriptions of tests and verifications to be implemented.

It is quite possible that the flight software can be standardized to the point of combining equations routines into fixed subprograms, and into fixed portions of program assemblages. If these assemblages are so organized, perhaps even fixed rope modules or ropes could be made for use in all later flights. The ideal situation would be to approach the operational mission with a complete flight program assemblage which has been flight proven. In this case the second step in software development would be reduced to only minor refinements in the flight program toward the end of the Apollo development program. These refinements would be minor adjustments required to satisfy the specific requirements peculiar to the flight test in preparation.

The complete operational software package may require thoughtful organization and composition of the equations to stay within the guidance computer memory capacity and timing constraints. Thus, a computer budget should be developed to allocate memory and guidance cycle portions to each subroutine or subassemblage, to guide the equation development process.

The flight equation testing should consist mainly of guided flight simulations with the AGC and LGC flight equations programmed in the scientific computer language. This testing will determine the ability of the equations to reproduce the reference mission, abort functions, and to perform trajectory control when perturbations are present. The AGC and LGC compatibility studies are concerned with determining if the developed flight equation subprograms can be successfully transformed into the flight computer language. This is accomplished by performing computer memory and timing requirement estimates, specifying major and minor computing cycles, and adopting algorithms compatible to the computer capability. When this has been completed, the MIT Design Review Board reviews the program checkout plan, the equation testing results, the compatibility studies results, and the flight software equation descriptions. A feedback from this board to these functions is provided for the re-testing or re-evaluation of the software if determined by the MIT board.

The MSC Software Design Review Board receives the results of all checkout and testing performed in Step 2. After careful study of these data the board will either issue a flight equation approval letter, or recommend additional testing of the software. In the former case, this directive permits the recommencement of the programming and coding for the AGC and LGC programs (Step 3). In the latter, iterations are performed on the equation testing and computer compatibility operations until satisfactory results are obtained for the board. All verification results and simulation descriptions are documented in detail and sent to the MSC Data Bank. This data bank is operated by the G&C Division at MSC and includes all documentation associated with the AGC and LGC software. It acts as the central documentation depot and disseminates this data to all interested parties. Since the software validation process is very complex and involves a large amount of documentation, it is extremely important that the data bank contain the latest information concerning the flight equations and testing programs. This will prevent redundancies and reworking of software elements and permit milestones to be achieved under the very tight Apollo schedule.

Coincident with the flight equation testing is the modification to the targeting and verification programs, micro simulations at MSC and MIT, and the real-time ground program. Current data from hardware systems checks and testing are used to update or modify the models used in these simulations. Information from qualification testing for previous flights is used in the targeting program changes. The final program configurations are documented and transmitted to the data bank.

During the software preparation operations in Step 2, the operational mission subprogram assemblages, to be used in the standardized library are being defined. Data from the MIT computer compatibility studies assist in the determination of the ultimate software requirements. This provides a functional interface with the current software and promotes a continual updating of the available flight routines for Apollo.

4.1.3 Programming and Checkout of AGC and LGC Flight Equations

In Step 3, the flight equations will be programmed and coded for the AGC and LGC using the equations described in the flight program assemblage document. Special programming techniques should be used where possible as a means of detecting computer arithmetic errors. These include self diagnostic tests, memory sum checks, reasonableness and limit checks for critical inputs or results, and sequence check routines to detect program jumps caused by transient malfunctions. The AGC/LGC Programming Manual, which is supplied by Raytheon, should contain programming rules and information on the flight computer input/output operations.

Program checkout of the flight equations is performed using the plan approved by the MSC review board in Step 2 with an interpretive simulation. This simulation uses an interpretative computer subroutine to permit manual checkout of the equations and "bench tests" by driving the particular program phases with representative mission profiles as inputs. All contingency program paths are executed to insure satisfactory performance.

At the completion of the program checkout, proper documentation will be prepared which will include a summary of the results of the checkout, and a description of the programmed equations and constants. A program symbolic listing and card deck is also generated at this time. It is used in targeting and qualification testing and copies are sent to NAA/S&ID and GAEC for use in modifying their hybrid simulations. This configuration of the AGC and LGC flight programs becomes the base line version from which all future program changes should be referred.

This documentation, as well as the Qualification Testing test plan is submitted to the MSC Software Design Review Board for approval. Upon acceptance of the data, the board issues a directive which releases the AGC and LGC programs for targeting and qualification testing. At the same time, all documents are sent to the MSC Data Bank for proper dissemination.

Modifications to the NAA/S&ID and GAEC simulations are made simultaneously with the computer programming and program checkout. The inputs are obtained from the analysis of previous tests performed using these simulations. It should be noted that this program updating is a continuing process throughout the validation process for all of the Apollo missions. A detailed description of the modified simulations is prepared and sent to the MSC Data Bank.

Since vehicle hardware integration and testing will be conducted concurrent to the software validation, the test results which affect simulation models will be properly implemented into the simulation programs. These results should be summarized in one document and become the source for all software simulation programs and targeting. This information is also used to upgrade the guidance ground program shown in Step 2.

4.1.4 AGC/LGC Program Targeting and Verification

It has been recommended that the flight software for the Apollo missions be developed in a modular form so that software changes which exist from flight to flight, excluding launch dependent parameters, can be implemented with a minimum of validation effort. This effort involves qualification testing of the new modules to evaluate their functional and interfacing properties. As a result of the current Apollo flight schedule, this software modular concept will introduce situations where it will only be necessary to determine the launch dependent constants when the trajectory and vehicle are identical for several flights. For these cases only a minimum amount of qualification testing will be required.

The targeting function in Step 4 should be performed on an engineering simulation program using a "hardware orientated" version of the flight equations i.e., using the flight program algorithms in the scientific program coding form. This will permit very close approximations to the results of the micro simulation and with much less computer running time. The targeting operation is governed by the approved targeting procedure from Step 2. All targeting constraints and revisions to the simulation models will have already been implemented into the targeting program from the data bank.

The targeting verification functions performed by MSC and MIT permit the mission and launch dependent constants to be evaluated over extreme operation conditions. It is desirable to have, in addition, an independent verification performed which will ensure that no errors have been overlooked and that all mission constraints are being observed This verification does impose the conditions that any changes to the flight programs be properly documented and transported to the proper agencies. It is only in this way that the verification processes can be considered valid. The verification plan used by the particular agencies will specify the tests to be performed according to their simulation capability.

Upon completion of the verification process, a critical comparison of the results is made. Considerations will be given to simulation configurations, accuracies, and operating conditions. Any discrepancies will be resolved by mutual re-testing. If this is unsuccessful, the problem area will be carefully documented and given to the MSC review board for further study.

The targeting verification test results and the flight constants listings are sent to the MIT Review Board for approval. At this time, the flight rope verification plans to be used by Raytheon and MIT are also received by the board.

In addition to the above documentation, the MSC review board receives the NAA/S&ID and GAEC simulation test plans and the real time ground program checkout plan. The final approved flight software is integrated into the ground program and flight rope fabrication begins.

4.1.5 AGC/LGC Program Qualification Testing

When new software is generated for the AGC or the LGC, it must thoroughly be evaluated to determine its performance capabilities. This is the function of Step 5. The qualification testing consists of micro-simulations of the flight equations together with the vehicle dynamics under all anticipated variations and extremes of vehicle performances, hardware tolerances, and mission environment. Detailed simulations of the vehicle dynamics and equipment operations such as available in the MSC general micro-simulation should be included only to the extent that is needed to determine the effect of the computer program on system operation.

In the qualification testing, open-loop response tests should be included to serve as an accurate engineering verification of the programming used for the AGC and LGC flight equations and constants. This subject is further discussed in Section 6.2. An MSC independent qualification testing is recommended in Step 5 to provide additional confidence in the guidance software. The test plans to be used will have been approved by the second review board in Step 2. Since more than one agency is doing this testing, close coordination of all vehicle, environment, and program data between the particular parties is required.

In Step 5, the testing results are compared and coordination is made with all interested agencies within MSC. If significant discrepancies in the testing cannot be reconciled, the MIT or MSC review board will be brought into the problem for corrective action. Since the software targeting effort and the qualification testing are performed almost simultaneously, one MIT and MSC review board, number 4, will evaluate both results. The reason for showing these boards in both Step 4 and 5 is to denote the simularity and individuality of these functions.

During this qualification testing period, NAA/S&ID and GAEC have been performing various software functional and interface testing on their hybrid simulations. These results, along with the qualification testing analysis, is examined by the MSC Board No. 4 and results in a program release certification for rope memory fabrication.

4.1.6 Rope Memory Generation and Hybrid Simulation Testing

In keeping with the principle of optimizing the software validation process by generating the guidance program in modular form, Step 6 permits some duplication of standard modules as well as the fabrication of revised modules, which may permit easier schedules. As shown in Step 6, the variable program (destruct) memory tape is generated from the binary program deck and can then be used directly in the MIT Hybrid testing.

To fabricate the AGC and LGC flight ropes, the guidance program is first converted into a punched tape for use in the memory wiring machine. At the same time a check tape is generated for use in the bit-by-bit verification of the flight ropes. All respective format requirements for each step should be documented to ensure proper tape preparation. The output of this operation provides a complete identification and description listing of the fixed and variable tapes. This listing remains with each generated tape and in the basic reference for the remainder of the validation process.

In Step 6, the punched tapes are compared to the binary program deck. Any discrepancies are corrected by iterating on this tape generation process. A total of two sets of wiring machines tapes and approximately three sets of each of the flight computer DRO tapes are generated per flight. One set of wiring machine tapes is used by Raytheon for fabrication of the fixed computer memories (ropes). The other set is used for bit-by-bit verification. This set contains one tape for the command module computer and one for the LEM computer. The appropriate documentation will be required here to specify the form of the wiring machine and rope checkout preparation processes.

One set each of the DRO tapes is sent to the ACE program, and the flight computers at KSC. MIT will receive a set of these tapes for use in their tape checkout tests and hybrid simulation runs. This set is ultimately sent to KSC to be used as a spare.

The number of DRO tapes and flight ropes required to support each flight will depend on the role played by the NAA/S&ID and the GAEC hybrid physical/digital simulations. The use of the full core rope simulator and memory card decks would be sufficient to verify the hardware modeling for software validation purposes. However, flight ropes might be needed to perform other functions with these simulations not connected with flight software validation. These requirements should be determined so that the number of ropes and DRO tapes required to support a flight can be fixed.

Approximately three sets each of command module computer flight ropes and LEM computer flight ropes should be fabricated by Raytheon and verified according to the defined test plan in Step 5. MIT and the Kennedy Space Center will receive one set of ropes for each computer, while the third set for each computer might go to NAA/S&ID and GAEC, if it is decided to do so.

The fabricated flight ropes, along with rope verification results documentation are sent to MIT for functional and interface testing using their hybrid simulator. Each rope set should be uniquely identified and contain the proper description documentation, only one set of ropes will be checked at MIT since the identity between these sets will have been established during the Raytheon verification operation.

The fifth MSC Software Design Review will consider the MIT rope checkout results and the preliminary functional testing by NAA/S&ID and GAEC. Since a MIT review board will have already analyzed and approved the MIT testing, this task should only require a minimal effort by the MSC board. The test procedure to be used in the G&N integration tests at the Kennedy Space Center (KSC) is reviewed by the board at this time.

This design review board will also review the results of the real time ground program checkout which has been performed concurrent to the MIT testing. In particular, the compatibility of the ground program with the flight software will be established. This includes a critical evaluation of the extensive interfaces between these two systems. When the review board approves the fabricated rope memories and DRO tapes, a directive is issued which certifies the software for flight and permits its integration with the guidance hardware at KSC.

4.1.7 Hardware Integration and Ground Testing

Two sets of certified flight ropes and two sets of DRO tapes, along with the proper descriptive documentation, are sent to the KSC for use in Step 7. The remaining flight ropes and tapes might be sent to NAA/S&ID and GAEC for final testing using the hybrid COSYDYVE and GAEC simulations. It is not necessary to send the AGC flight ropes and tapes to GAEC since their primary area of interest is the LEM guidance program.

Upon completion of the G&N assembly and integration test, loading of the DRO tapes into the ACE equipment, and incorporating the tapes and ropes into the flight computers, the results of these operations are briefly reported and sent to the Flight Readiness Committee for review. These reviews should be conducted by NASA and contractor engineers at KSC. This committee is responsible for certifying that all preflight operations and ground testing has been satisfactorily accomplished. They will also examine the results of any final hybrid testing by NAA/S&ID and GAEC, if these tests apply. This committee should contain members from MSC, MSFC, and the major contractors. A member of MSC should chair this committee. This committee should review the documented results of all major tests performed in the ground checkout phase. Since this is the final series of tests before the flight, all discrepancies and failures in a test will be resolved before proceeding to the next test. This process is repeated until the committee issues a release to the Launch Control Center which releases the program for launch countdown.

The ground testing shown in Step 7 will be of approximately three months duration. All test should be planned in detail and properly documented. Since this will be the final assembly and integration the complete set of Apollo software and hardware, adequate testing should be provided for a functional checkout under the nominal and abort conditions. All astronaut override capabilities and interfacing should be thoroughly tested on a subsystem and systems level. The ground test program should provide for at least one condensed mission rehearsal. The nucleus for the ground test program currently exists in the Ground Operations Requirements and Plan (GORP) series.

Throughout all tests and checks performed in Step 7, MSC and all concerned agencies will monitor the progress and all test results. This will ensure that the results are compatible with the overall mission objectives before the software programs are released for flight.

4.1.8 Launch Countdown and Flight Operations

The software validation process continues during the launch countdown phase by introducing automated check routines into the programs. The AGC and LGC can be monitored through the automatic checkout equipment (ACE) for determining satisfactory operations of the flight computers.

During the flight, the verification process continues by exercising various memory sum checks, sequencing checks, reasonableness tests, and diagnostic routines. Astronaut functional and interfacing checks with the G&N system will be monitored by displays and telemetry. The real time ground program should be performing similar verifications based upon data from the down-link system. In addition to the pre-programmed checks, both programs should contain the capability to perform testing using data input from external source such as astronaut or a launch operations officer.

4.1.9 Post Flight Analysis

The "quick-look" analysis is concerned with determining the overall equipment and software performance during flight. Since the report on the analysis is required shortly after the flight (usually one or two days), it is not possible to obtain a detailed analysis. The software performance is reconstructed using the interpretative computer or micro simulation and the precomputed nominal trajectory. The tracking and telemetry data are reduced and the results compared to recover the instrument and propulsion performance, the environments experienced, and to assess and explain any malfunctions that were observed.

The "fine-grain" analysis in step 9 is a more detailed investigation into the software and hardware performance during the flight. The reconstructed best estimate trajectories can be used since time will be available for proper reduction and data estimation. This provides a more realistic environment for checking the operations of the AGC and LGC programs. This capability imposes a requirement for adequate telemetry coverage during the flight so that all malfunctions can be properly monitored. Various techniques will be used in this analysis to translate and checkout modifications to the software which might prevent in future flights any of the observed malfunctions. This is an important step in the validation process since refinement of the existing software should include the experiences gained from actual use in an operational environment.

The Post Flight Analysis Committee correlates and coordinates the various post-flight analysis performed by each agency and contractor. The chairman should be a member of MSC. The committee should contain representatives from all agencies participating in the post-flight analysis. Procedures should be defined by this panel in such areas as telemetry requirements, data processing techniques, and distribution of analysis responsibility.

4.1.10 Software Change Procedure

Throughout the entire software validation process it may become necessary to insert changes to the AGC and LGC program. Until Step 6, where the rope memories are fabricated, the flight program will exist in the basic form of a binary card deck. Therefore, any required software changes can be implemented with a minimum of effort but it is necessary to perform re-targeting and verify the change into the program. However, once the program is "frozen" by rope fabrication, the operations involved in making any software changes require longer lead time and may cause launch schedule slip. Thus, a realistic plan must be mechanized for implementing necessary software changes after the program has been finalized. A recommended procedure is shown in Figure 10.

Changes to the flight software can originate from two primary sources. First, an unexpected result can occur from software or hardware tests which can be attributed to an error in the software logic, or which could be effectively resolved by modifying the software. Second, a mission, procedural or equipment change can arise during the software development process which affects the software. The first situation is denoted at point (A) in Figure 10 and the second at (B). The change process is initially different for the first source. This is required since the suspected error might be resolved quickly, as in the case of an error in the sign of a guidance constant, or a physical defect in a memory rope. A rapid software correction path has been provided in the procedure to handle these situations. MIT is primarily responsible for this type of change with the intended action and subsequent progress coordinated with MSC.

For more involved changes, a longer, more detailed change procedure is required. The first step is concerned with making a detailed simulation analysis of the problem area. All dependent agencies and functions will be used, when necessary, to aid in determining the error source. In particular if the fault can be attributed to an astronaut software interface, rigorous coordination between the two functions will commence to attempt a fast and definite solution to the problem.

The MSC Software Design Review Board should hold combined meetings with the Change Control Board (CCB) for the sake of expedience which is usually associated with such program changes. These meetings will review the results of the analysis connected with the hardware or software error and recommend the type of correction to be made. The MSC review board will be of significant value to this task since the CCB members will not necessarily be knowledgable in all aspects of the software. When approved, a change directive will be issued by the CCB board and corrective action by the particular agency will begin. After the re-testing has been satisfactorily completed, both the MIT and MSC review boards will review the results. A software change approval directive finalizes the correction and the appropriate NDRO and DRO tapes are generated. These tapes are then used for implementing the software change into the particular flight computer.

4.2 CONTINGENCY PROCEDURE

The contingency procedure which deals specifically with planned alternate flight configurations is not to be confused with the change procedure which deals with unforeseen software modifications brought about due to changes in mission procedures, hardware modifications, or faults in the rope equations and logic.

In the event that an alternate or contingent flight plan must be employed due to unsatisfactory performance of preceding flights, preparations for these contingent flights must be made sometimes well in advance of the launch date since this decision point may occur as late as a month before launch. The replacement ropes which would be used are similar to those associated with previous flights but may not be similar to the ropes used in the primary mission computer memory. Because the possibility exists that these ropes may bypass some of the ground tests if the interchange of ropes occurs shortly before launch, special testing of these ropes must be performed and completed prior to the time they are needed. The problem then is to determine the best procedure for phasing in revised flight ropes in the last few months before launch without slipping the flight or affecting the integrity of the ground tests. These tests should be conducted with flight qualified hardware and may be a condensed version of the ground tests since many of these tests may either be duplicated or not affected by the contents of the ropes.

A contingency procedure as shown in the logic flow diagram Figure 4-12 is recommended. The contingency program should be initiated as soon as the difference in the two ropes can be identified and their affect on the ground testing defined. This will determine what special ground tests are required before the ropes are replaced at the time of flight decision.

The primary flight rope program should be conducted to the attainment of rope release even if the decision point to employ a contingency configuration occurs before the ropes are released since these ropes, in turn, may become contingency flight ropes for subsequent missions. The decision point, therefore, has very little impact in the primary flight rope procedures other than to instigate the replacement of ropes. In the contingency procedure, however, if the decision point occurs before or during the special ground tests, it may be desirable to discontinue these tests and to use the replacement ropes as soon as possible in the ground test program.

The magnitude of the potential problem discussed here is dependent on the similarity of the test portions of the computer ropes and the dependency of the ground test programs on the specific flight rope contents.

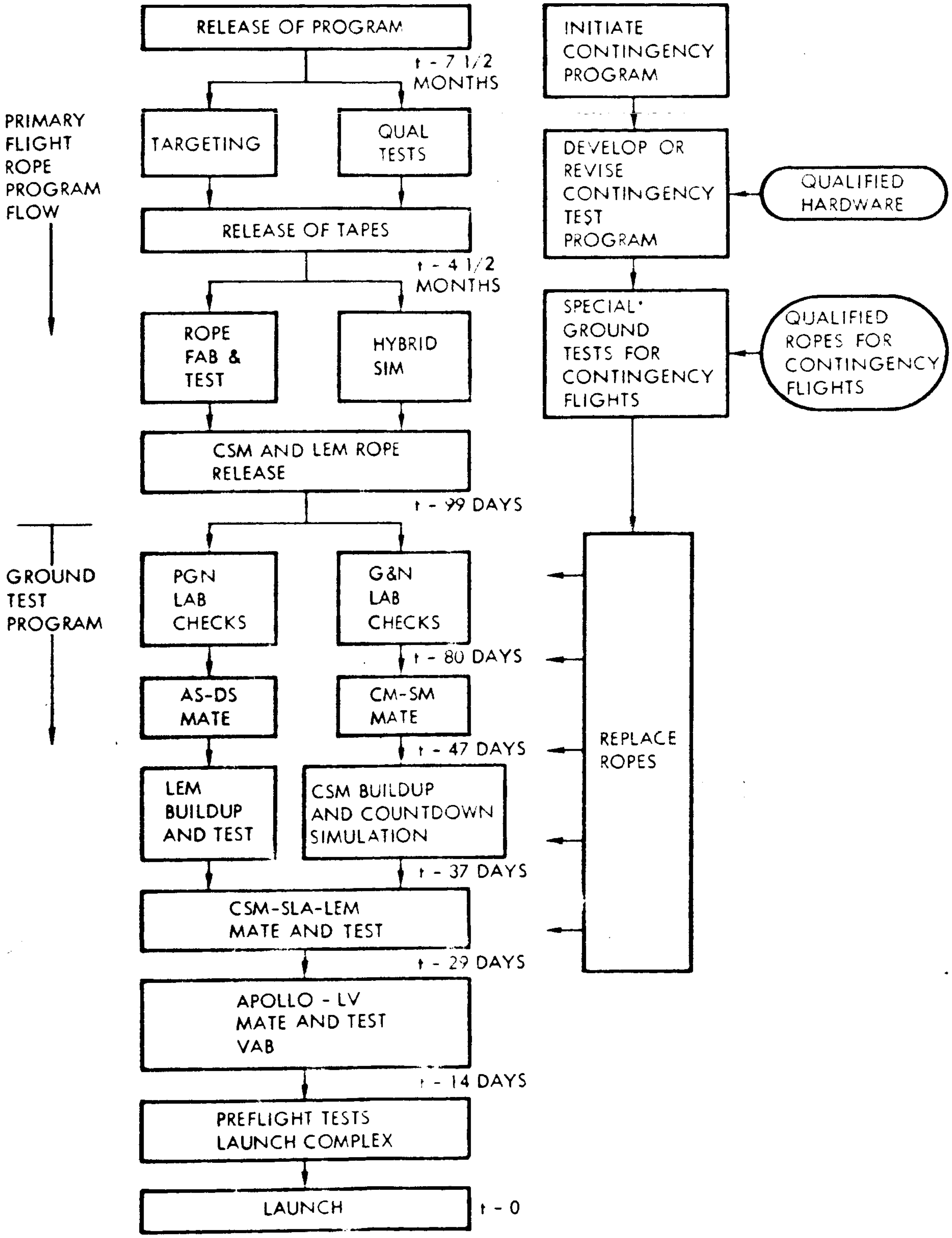


Figure 4-12. Contingency Procedural ~~Procurement~~ Flow Diagram

Obviously, if the test ropes are identical between the nominal ropes and the contingency ropes and if the ground tests results are independent of the internal computer flight programs then a switch can be made at any point in the ground testing without much special testing. However, it is quite likely that the nominal mission ropes and the contingency mission ropes will contain differences, which implies some amount of special testing on the side before the ropes are replaced and the ground testing continued.

4.3 FLIGHT SOFTWARE INTERFACE SUMMARY

In this report, the validation of the flight software has received the major emphasis. However, it is obvious that the validity of the software depends on its consistency with many other facets of the Apollo program. This plan has made provision for the coordinating of the flight software interfaces but in some cases these have not been clearly distinguished. In this section, the general interface areas will be summarized and the method of interface verification identified. This will help to distinguish the steps in the software validation procedure primarily concerned with the interface of interest.

The major areas of flight software interface are:

- Spacecraft G&N equipment
- Spacecraft propulsion and altitude control systems
- Telemetry link
- Crew operating procedures
- Saturn booster guidance system
- Guidance system oriented ground support equipment
- Real time ground program and mission control procedures
- Mission design and related documentation
- Software description documentation
- Test ropes and ground checkout

The compatibility of the flight software with the spacecraft G&N equipment is insured first through the accurate modeling of these components and subsystems in the hardware oriented flight simulation programs used in qualification testing. This modeling relies on maintaining a current and accurate Handbook of Vehicle and Equipment Simulation Models document. These models are further justified in the MIT hydrid simulation rope verification program where the rope/computer interface is functionally tested along with the rest of the G&N equipment.

The software interface with the spacecraft propulsion and attitude control systems is also tested during software qualification tests using accurate simulation models controlled by the Handbook of Vehicle and Equipment Simulation Models. These models are verified by means of the NAA/S&ID and GAEC hydrid simulation tests. Special tests are also performed during equation tests or qualification tests to ensure a compatible stability margin between the guidance and control functions.

The consistancy of the flight software with the telemetry link is fundamentally defined in the Software Specification Document and thus becomes an integral part of program checkout and qualification testing. This functional interface is later physically checked during the ground checkout procedure.

The compatibility of the flight software with the crew operating procedures if first defined as part of the Software Specification. It is later checked during program checkout and qualification testing by either pre-programming the astronaut control of the computer operation or providing an engineer to test the man-in-the-loop aspects of the astronaut/computer procedures. Once the compatibility of the software with these procedures are verified, the mission simulators are used to train the astronauts in the use of these procedures. Any astronaut aids in the form of written data carried onboard the spacecraft should also be reviewed for consistency with the software during software qualification testing and targeting verification.

The agreement between the spacecraft software and the Saturn booster guidance computations is confirmed in the Programmed Equations and Constants Document, the Equation Test Results Document, and the Qualification Test Results Document and in the targeting verification operation. Since this interface will be in the form of monitoring during early flights, the criteria used in the comparison computation of the real time ground program must be checked against the qualification test results.

The ground support equipment interface with the flight software is mainly through the ACE computer program. The computer load and verify operations will be checked as part of the ground testing phase, particularly in the G&N system integration and assembly. However, any monitoring or interpretation of the flight software by the ACE program will have to be verified by either special studies or by definitions of constraints in the Software Specification and by specific qualification tests directed to this interface. The interface of the flight software with the real time ground program and mission control procedures is a very complex one. The extent to which the software must be simulated in the mission control center has not been determined. However, this interface is recognized in the recommended procedures by including the real time ground program intimately in the software review process in several places. The more significant review is before rope release to ensure that the final version of the flight software is reflected in the mission control simulations. It is also important that the same vehicle and equipment models and targeting data are used consistently in both sides of the interface as provided in the procedures.

The mission design considerations are incorporated in the flight software in the flight software specification and in the targeting requirements documents. The verification of this interface is provided in the equation tests, the qualification tests and in the targeting verification. On the other side, the final mission planning is performed using guided flight simulations reflecting the flight software and targeted constants.

Since the flight software is used by many organizations involved in the Apollo mission preparation, it is imperative that the descriptions of the software available in documents be complete and reliable. This will not only provide the diverse users with accurate information, but will provide a means of "trouble shooting" anticipated software problems during various tests in which the software participates and in post flight analysis. These documents also serve to define the software configuration for the purposes of control. The procedure identifies the significant portions of the software which warrants specific documented descriptions.

The flight software/test rope interface becomes significant because the test rope is used to checkout the computer/hardware functions and may take the place of the flight rope during certain phases of ground assembly and test. If the test ropes and flight ropes are not consistent in terms of the methods of operation on computer input/output signals, then equipment tests performed using the test rope may become invalid or that flight rope faults may bypass certain equipment tests. This interface is verified mainly with the MIT hydrid simulations using flight ropes and during ground testing involving both the test and flight ropes.

Because of the complexity of the Apollo program, and the relatively limited scope of this study it is possible that all software interfaces have not been treated specifically in this validation plan. However, if these omissions are identified, they can be incorporated within the procedure outlined here, and the incorporation of specific tests in the test plans designed to verify each interface.

5. FLIGHT SOFTWARE VALIDATION DOCUMENTATION

This section describes the necessary documentation for complete preparation, review, and testing of the Apollo AGC and LGC flight software. This documentation forms an integral part of the flight software verification process for the anticipated nominal Apollo mission. Its particular function in this process is in establishing a control over the very large and complex software program. This is accomplished by requiring that the demonstrated performance of the software as defined meet the requirements and constraints which are specified in the particular documents. The MIT and MSC review boards incur the responsibility of determining that this does occur.

The required documents have been grouped together in a form which is designed to minimize the effort required for their preparation and revision. By providing one comprehensive test plan document organized in replacable sections, the effort required to incorporate revised test plans or to expand on previous tests is reduced. Only the Software Specifications, the Test Plan, selected software description documents require the formal approval of MSC. Many of the required documents, or sections of them, can remain unchanged over a number of flights, and would only require revision when a major change in the form of the software is required,such as the incorporation of the Block II computer.

The recommended software validation documents are grouped into the following six categories:

- Specifications
- Test Plans
- Test Results
- Software Descriptions
- Approvals
- Handbook and manuals.

5.1 SPECIFICATIONS

The Software Specification is produced after the specific mission software requirements have been defined in Step 2, and it forms the basis for all future testing of the AGC and LGC flight software. These specifications will include software requirements under nominal and non-nominal operating conditions, determination of guidance backup modes, and a specification of the perturbations to be mechanized for non-nominal performance studies. Also includes are specifications of the vehicle, IMU, and computer simulation models. All astronaut/software interface operations and procedures should be included. The specification should define quantitative software performance criteria so that the software test results will be easy to interprete and approval can be made straight forward. This document should also contain all vehicle and equipment performance criteria to be used in determining the operational performance capability. This document is used to determine the method and objectives of all test plans used in the verification process.

5.2 TEST PLANS

The contents of the Test Plan are shown in Table 1. Many of these plans will remain unchanged over a duration of many flights. This would occur if the tests are comprehensive and generally applicable and the software becomes standardized. When modifications are made for a specific flight, it will be necessary to perform only the functional and interface testing on the new routines developed or the new combination of several standardized subroutines.

5.2.1 Flight Equation Test Plan

This plan is used for checking out the preliminary AGC and LGC flight equations developed by MIT. It will contain tests which use the engineering or ideal simulations of the equations in a closed-loop guidance configuration. The tests should be designed to demonstrate the mission performance under both nominal and non-nominal conditions. They should also be capable of completely checking all proposed individual subroutines. This task can be considered to be the basis for the software performance analysis which is part of the guidance equation specification. This test plan should be revised only when it is necessary to perform special testing on the modified flight software.

5.2.2 Program Checkout Plan

This plan defines the early tests or the coded form of the equations when programmed for the guidance computer. These tests should include those which are used when running "bench test" or interpretive computer simulations (ICS) which utilize the micro simulation program. This plan should also include tests which demonstrate that the flight program is consistent with all of the computer logic and arithmetic operations. This document need only be written once, and modified only when new guidance program subroutines become available. A further description of the program checkout process is given in Section 7.2.

5.2.3 Software Qualification Test Plan

This test plan describes the qualification testing which is used to demonstrate that the flight software when included as part of the G&N subsystem is capable of guiding and controlling the vehicle in all of the operating modes for the mission. These tests use an ICS/FS simulation, which consists of the micro simulation where the loop has been closed between the flight computer and the vehicle for all anticipated mission conditions. Also simulated are the detailed models of the flight hardware which interfaces with each of the flight computers. This test plan should also provide tests in which all of the nominal and abortive astronaut interfaces are rigorously exercised. The outputs of the computer computations which are required for mission control and monitoring should be checked for completeness. The limits to the performance of the AGC and LGC software should be explored until all nominal, abortive and contingent operation modes are demonstrated satisfactorily.

An important part of this plan is the specification of that portion of the qualification testing which should be performed by MIT and MSC considering the unique capabilities of each simulation program. The division of responsibility should be partially complementary, but it is important to specify some identical runs for the purpose of verifying the simulations themselves.

The Software Qualification Test Plan should be modified to reflect changes in the software, vehicle configurations, or updated flight environmental information, mission and equipment performance. The revisions to the plan should include description of the changes, new equipment, or simulation requirements, and functional and interface operation procedures. Subsection 7.3 contains a further description of the software qualification process.

5.2.4 MIT Flight-Rope Verification Plan

The testing to be performed by MIT on the AGC and LGC flight ropes when they are received from Raytheon should be described in this plan. Tests to verify the functional characteristics of these ropes, and their interface with the G&N equipment with the MIT physical AGC-LGC/digital simulation are defined. All G&N equipment to be used in this simulation should be specified in this plan. The acceptable limits and deviations for all interface testing should be specified. This verification phase will be used to complement the tests performed on the ropes by Raytheon.

This plan would only have to be revised when equipment interfaces or configurations change significantly.

5.2.5 Raytheon Rope-Verification Plan

This verification plan primarily should consist of making a bit-by-bit comparison check of every fabricated flight rope with the check tape. This process should be completely automated, and able to indicate any discrepancies when they occur. This plan should be reviewed by the MIT Review Board prior to its initial use. This plan will not require modifications between flights, after it is approved by MIT and MSC.

5.2.6 NAA/S & ID Simulation Test Plan

This describes the series of planned tests to be performed with the flight software in the NAA/S&ID hybrid physical simulation. All information concerning the procedures to be used in each test, flight hardware to be employed or simulated, and the necessary inputs and desired outputs should be completely described in this document. This test plan should also include functional and interface testing using the flight hardware. It may require revision to specifically verify spacecraft equipment changes. The MSC Review Board should examine this document to ensure that the proposed testing is compatbile with the overall flight software validation process.

5.2.7 GAEC Simulation Test Plan

This test plan should be analogous to the plan in the above subsection. The MSC Review Board will be responsible for determining that both hybrid simulations are being used in an optimum manner individually. Redundant testing should be kept to a minimum and be used only to verify the functional performance of the CSM/LEM equipment interface. It should be modified similarly to the NAA/S&ID test plan.

5.2.8 Real-Time Ground Program Checkout Plan

This checkout plan is used to establish the capability of this real-time ground function to effectively perform the required flight computer functions during the simulated flights. All data interfacing with the telecommunications system should also be tested. The required equipment and inputs and outputs for each test should be defined. This document should be coordinated with the software program checkout and qualification testing processes. This will promote standardization of the checkout procedure and help minimize the change requirements.

5.2.9 G & N Integration Test Plan

This test plan is used for integrating the G&N equipment together on a system and component level at the Kennedy Space Center. Since the command module and the LEM primary guidance system are of the same basic configuration, one plan could be prepared to incorporate both systems.

This plan should include a complete functional checkout of both of the flight systems. Portions of this test procedure already exist as part of the GORP series of checkout testing. Thus, minor modifications should only be required to adapt this plan to both flight systems.

5.2.10 Vehicle Assembly and Integration Test Plan

This document is described only functionally in this validation process, since it already exists as part of the GORP series of checkout tests. It should be noted, however, that the test results must be provided in a form which can be easily checked against the performance criteria. This permits the respective review boards to detect anomalous behavior in the flight software without resorting to detailed analysis.

This procedure should remain fixed for a given guidance system configuration. It should be modified only when the change from Block I to Block II is made.

5.3 TEST RESULTS

The documentation necessary for reporting the results of all testing during the software validation process is listed in Table 5-2 and described below.

The test results reports are basically informal documents which are concerned with presenting these results in a concise and accurate form to the MSC and MIT Review Boards. As such, these reports (or memorandums) are not approved documents. The degree of informality to be used in presenting the test results will depend upon the nature of the involved test. For example, the flight-rope verification test results might consist of a few pages which state that all test objectives have been achieved without reporting details of each step. However, when the test involves checking the performance of various parameters in the software subroutines, for example, (which might be the situation during qualification testing) it would be necessary to be more specific in presenting the test results.

It should be necessary to specify the format to be used in each type of test result only once. This standardization of reporting each type of result will provide a rapid means of communicating the status of the flight software between the testing agency and the responsible review board.

5.3.1 Flight Equation Testing Results Summary

The results of MIT work performed on the preliminary guidance equations,using an engineering-type simulation, should be summarized and documented. It should include an indication of the expected level of performance on the software. All positions of the guidance logic which produced marginal or unsatisfactory results should be delineated.

5.3.2 Program Checkout Results Summary

These results are a summary of the test results obtained from the program checkout function using the approved test plan. Any modifications to this checkout plan,which might result from software modifications for a particular flight, should be documented in this report. Reference should be made here to the programmed equations and constants used in the checkout. This summary should also describe any functional change made to the flight programs during the testing and explicit reasons for doing so.

5.3.3 Qualification Test Results Summary

This summary document will contain the results of the software qualification testing performed by MSC and MIT. This document should reference the most current description of the flight software. It should specifically compare the results obtained against the software requirements specification.

5.3.4 Targeting Verification Test Results Summary

The results of the targeting verification process by the two principal agencies should be properly combined and documented. This test results summary should include, in addition to nominal mission performance data, the results of the automated test performed to exercise the software in the presence of the selected perturbations used and the abort mode tests.

5.3.5 Hybrid Simulations Test Results Summary

This document contains the results of the nybrid simulation tests performed at NAA or GAEC using the flight program decks and DRO tapes. These results should be used to determine the validity of the equipment models used in software simulation programs.

5.3.6 Flight-Rope Verification

This report contains the results of both the Raytheon and MIT rope verification tests. The Raytheon tests consist of a bit-by-bit comparison of the fabricated flight-core ropes with the check tapes. A separate verification data sheet should exist for each manufactured rope.

Since it will not be necessary for MIT to thoroughly check every flight rope received from Raytheon, their test verification will represent several sets of ropes, i.e., three flight ropes per set, whereas the actual testing might be done with only one complete set. This is possible since Raytheon will have performed rigorous testing on all of the ropes prior to delivery to MIT.

5.3.7 Real-Time Ground Program Test Results Summary

This document summarizes the results of the real-time ground program checkout testing. It is useful to the software validation process by determining the effective interface of the real-time ground program with the software. It is also used to establish the compatibility of the ground software function to the flight software.

5.3.8 Vehicle Assembly and Integration Test Results Summary

This summary contains both the G & N integration test results for the AGE and LGE, and the vehicle assembly test results. The former results will be used to establish confidence in the integration of the guidance software with the guidance hardware. Since the tests will be on both a component and systems level, this summary will form the basis for further systems integration studies during the vehicle assembly and checkout testing phase at KSC.

The vehicle assembly test reports are brief descriptions of the test results for each step in this test program. They are presented to the Flight Readiness Committee for use in approving the results of the ground checkout procedure.

5.3.9 "Quick-Look" Post-Flight Analysis Report

The purpose of this report is to produce an immediate indication of the performance of the flight. It should identify any obvious anomalies in the data and indicate the gross performance level of the vehicle systems. This will generally be a separate report coordinated between those agencies participating in the post-flight analysis.

5.3.10 "Fine-Grain" Post-Flight Analysis Report

This report encompasses the results of the fine-grain post-flight analysis of the radar data. It contains the detailed results of the flight test. This includes reconstructed interpretive simulations (micro) of the software. It provides insight into potential problems in the software and is the basis for future refinement. It is also a separate coordinated report.

5.4 SOFTWARE DESCRIPTIONS

This group of documentation describes the flight software assemblages, flight-memory contents, and the major software validation simulations which are required in the verification process. A list of these contents is given in Table 5-3.

The only controlled document requiring approval in this group is the Flight Equations Description Document which may be revised to reflect the programmed equations and constants after program checkout is completed. Most of the other documents are informal and are for information purposes.

5.4.1 Flight Equations Description

This document should describe all of the AGC and LGC flight programs assemblage used on a particular mission. The flight program assemblages are made up of the subprograms from the "Sunrise" and "Corona" series library at MIT, plus any special functions developed in Step 2 of the procedure. The document is intended to provide those working with the software to understand its philosophy.

5.4.2 AGC/LGC Programmed Equations and Constants

Upon completion of the program checkout of the flight software for the command module computer and the LEM computer, the guidance equations (along with any modifications) and all constants are documented. This programmed guidance equation description should contain the complete flight program flow charts, much of which is concerned with logic other than guidance equations. The range of values of the variables (used for program scaling) and a set of preliminary constants, which are used for testing, should be included. All problems which were applicable to programming, i.e., timing requirements to account for interrupt, should be described, as well as the telemetry formats provided.

5.4.3 AGC/LGC Program Listings and Card Decks

This data package is a cover letter which uniquely identifies the enclosures when transmitted from MIT to other using organizations. It contains the symbolic listing of the AGC and LGC flight program and a binary card deck of each program. These are issued after the coded form of the flight program has been satisfactorily checked out on the interpretive computer simulation in Step 3. This information is used to establish the software configuration during targeting and qualification testing.

5.4.4 Revised Sunrise and Corona Series Library Description

This document should contain complete and detailed descriptions of all the new and modified MIT subprogram assemblages. It should be the source of all standardized subroutines. The test rope routines are also contained in this document. It is also useful as reference information for the program checkout operation in Step 2.

5.4.5 Target program Description

This report is continuously revised as necessary to reflect the current capabilities and techniques used in the targeting operation. This specification will include the technical requirements for the targeting program, descriptions of the general modes of operation and computation flow, and a description of the major sub-routines used and their interfacing requirements. Also included are descriptions of the required outputs, tests, and verifications tests to be used.

5.4.6 Wiring Machine and DRO Format Description

The wiring machine and DRO format document should be prepared by Raytheon to provide MIT with a description of the format required of the punched tape to be loaded into core rope wiring machine. This document will permit MIT to directly generate the inputs to the core rope wiring machine. The wiring instructions furnished by MIT shall be of the proper format and medium (magnetic tape, punched tape or card deck) so that Raytheon is not required to modify the punched tapes but can feed them directly to the wiring machine.

This document will include the format and requirements needed for checking tapes on a bit-by-bit comparison to the rope readout if a difference exists from the machine input.

A description of the DRO memory format is also given which has been coordinated with the agencies responsible for the GSE and ACE. This insures complete compatability of the software to the associated checkout equipment.

5.4.7 Identification of Flight Ropes and DRO Tapes

This document is actually in the form of a cover letter which accompanies the transmission of the flight ropes and variable memory tapes. This document uniquely identifies the enclosed when delivered to using organizations.

5.4.8 Micro Simulation Descriptions

A description of the interpretative or (micro) simulations available at both MIT and MSC should be provided to the review boards to permit the evaluation of results and to define tests more explicitly. This description should contain all operating modes, limitations of the program, and the necessary input and outputs obtainable. The program capabilities to simulate the astronaut functions and its interfaces should be properly documented.

This document should contain a description of the level of modeling of the vehicle, environment, and equipment associated with the software. This simulation description will be helpful to the design review boards at both MIT and MSC in evaluating the software test results.

5.4.9 Targeting Data Package Description

This document contains all of the final guidance and navigation constants that were generated during the targeting procedure for the AGC flight program. This includes launch-time-dependent and mission-dependent constants, i.e., those constants which allow the mission objectives to be satisfied for the given vehicle and trajectory constraints. The flight-constants deck will be in the form of a set of IBM cards and a decimal listing. It provides additional plots and tables representing the results of the targeting operations such as firing tables, equipment setup, and the variation of selected variables throughout each launch window.

Since both MSC and MIT are involved in the software qualification testing, this document is a reliable means for quickly transforming the results of the targeting efforts for qualification testing.

5.4.10 Real Time Ground Program Description

This document primarily defines the functional description of the real-time ground program to be used on all Apollo flights. It contains a description of all computation used in processing tracking and telemetry data flight software, contingency and abort criteria status parameter descriptions and simular details. It also contains overall flow diagrams to understand the basic operation of the program.

5.5 APPROVALS

Throughout the software validation process, for a given flight, the MSC Design Review Board issues approval letters signifying that the flight software has satisfactorily passed each preparation step. These letters are recommended for the following steps.

a) Flight Equations Approval

b) Program Checkout Approval

c) Program Release for Fabrication

d) Certification of Fixed-Memory Ropes and Variable-Memory Tapes.

Certification letter a) is written by the MSC Software Design Review Board upon satisfactory completion of the flight equation development and testing in Step 2. The approval letter in b) signifies that the coded flight programs have satisfied all test objectives in the program checkout phase Approval letter c) is written when the flight programs have successfully completed all targeting and qualification testing, program certification. This permits the fabrication of the flight ropes for the AGC and LGC by Raytheon. The rope memories and variable-memory tapes are released by certification d) when the board has determined that the planned verification tests have been successfully passed.

In addition to the above approvals, the Flight Readiness Committee indicates their release of the flight vehicle to the launch operations direction with a brief certification.

5.6 HANDBOOK AND MANUALS

5.6.1 Handbook of Vehicle and Equipment Simulation Models

The purpose of this document is to compile in one volume all significant characteristics of the equipment and vehicle which are required in the simulation models. This will ensure that the simulations contain up-to-date information on the vehicle and equipment.

This document will be prepared by MSC and revised periodically to reflect all systems and subsystems changes and performance obtained from subsystem design verification and integration testing, for example.

Reference should be indicated, where needed, to the sources of the data and the other documents containing more detailed data. However, it should contain as a minimum, the definitions of all models required in qualification testing and targeting.

5.6.2 AGC/LGC Programming Manual

This document is provided by Raytheon to aid in the programming of the guidance computers. It lists the instruction capabilities, computational speeds, special features and subroutines developed, input/output preparation requirements and other similar information.

Table 5-1. Recommended Test Plan Summary

DOCUMENT	Originating Agency	Primary Receiving Agency	Availability in Months Before Launch	Figure Number
1. Flight Equation Test Plan	MIT	MSC	-15	4-3
2. Program Checkout Plan	MIT	MSC	-11	4-4
3. Software Qualification Test Plan	MIT	MSC	- 9	4-6
4. MIT Flight Rope Verification Plan	MIT	MSC	4	4-5
5. Raytheon Rope Verification Plan	Raytheon	MIT, MSC	- 5	4-5
6. NAA/S&ID Simulation Test Plan	NAA/S&ID	MIT, MSC	- 5	4-5
7. GAEC Simulation Test Plan	GAEC	MIT, MSC	- 5	4-5
8. Real Time Program Checkout Plan	MSC	MSC	- 6	4-5
9. AGE/LGE Integration Test Plan	NAA/S&ID	MIT, MSC	- 4	4-7
10. Vehicle Assembly and Integration Test Plan	MSC	MSC	- 4	4-7

Table 2. Test Result Documentation Summary

Test Result Summary	Originating Agency	Receiving Agency	Figure Number
1. Flight Equations	MIT	MSC	4-2
2. Program Checkout	MIT	MSC	4-3
3. Qualification Testing	MIT, MSC	MIT, MSC	4-5
4. Targeting Verification	MIT, MSC	MIT, MSC	4-4
5. Hybrid Physical Simulations	NAA, GAEC	MIT, MSC	4-5
6. Flight Rope Verification	Raytheon, MIT	MIT, MSC	4-6
7. Real Time Ground Program	IBM, MSC	MSC	4-7
8. Vehicle Assembly and Integration	NAA/GAEC	MSC/ Contractors	4-8
9. "Quick-Look Post Flight Analysis	MSC/ Contractors	MSC/ Contractors	4-10
10. Post Flight Analysis	MSC/ Contractors	MSC/ Contractors	4-10

Table 3. Software Description Documents Summary

Description	Originating Agency	Receiving Agency	Figure Number
1. Flight Equations	MIT	MSC, NAA, GAEC	4-3
2. AGC/LGC Programmed Equations and Constants	MIT	MSC, NAA, GAEC	4-4
3. AGC/LGC Program Listings and Card Decks	MIT	MSC, NAA, GAEC	4-4
4. Revised Sunrise and Corona Series Library	MIT	MSC, NAA, GAEC	4-3
5. Target Program	MIT	MSC, NAA, GAEC	4-3
6. Wiring Machine and DRO Format	Raytheon	MIT, MSC	4-7
7. Identification of Flight Rope and DRO Tape	MIT	MSC, ACSP	4-7
8. Interpretive and Micro Simulations	MIT, MSC	MSC, MIT	4-3
9. Targeting Data Package	MIT	MSC, NAA, GAEC	4-5
10. Real Time Ground Program	MSC. IBM	MSC, NAA, GAEC	4-7

6. SIMULATIONS AND PROGRAMS

This section contains a brief description of the major computer programs considered necessary for the software validation process. A summary of these programs is given in Table 6-1 including the agency performing the simulation or program, approximate availability schedules relative to the launch date, and the flow chart figure number where it is cited.

The remainder of this section contains descriptions of the simulation programs as applied to the software validation plan. These descriptions are only intended to clarify the functional description of the programs or to provide comments concerning their application. In a few instances, recommendations are made to provide modes of operation of the same basic simulation to perform various functions.

6.1 ENGINEERING GUIDED FLIGHT SIMULATION

The primary purpose of an engineering guided flight simulation is for use in the design, development and performance analysis of the early versions of the flight equations by MIT. In addition, a similar version of this simulation is used in trajectory shaping and mission analysis by MSC.

The principle sections of the simulation consist of mathematical models of the vehicle, aerodynamics, flight environment, rotational dynamics, a low frequency model of the vehicle controls system, and translational equations of motion as well as the flight equations. When it is desired to study only the guidance equations, scientific computer running time can be saved if a point mass version of the vehicle dynamics and a control system model which has a unity transfer function are used The flight equations are programmed in an idealized fashion taking full advantage of the scientific computer's instruction capability and word size. When used in the full six degree of freedom mode, it can be used to determine the effect of simulation model simplifications on software performance.

Table 6-1. Summary of Major Simulation and Programs

Simulation and Programs	Originating Agency	Availability in Mo. Before Launch	Figure Numbers
1. Engineering Guided Flight Simulation	MIT	-15	4-3
2. MIT Interpretive Computer Simulation	MIT	-10	4-5
3. MSC Micro-Simulation	MSC	-10	4-5
4. Hybrid (Analog/Digital) 6 degrees of freedom	MIT	-15	4-3
5. Hybrid/Physical Simulation	MIT	-11	4-4
6. Magnetic Tape to Card or Punched Tape Generator and Comparitor	MIT	-6	4-7
7. NAA/S&ID Hybrid/Physical Simulation	NAA/S&ID	-6	4-7
8. Wiring Machine Programs	Raytheon	-5	4-7
9. Memory Reader and Check Tape	Raytheon	-5	4-7
10. DRO Tape Read-In; AGC/LGC Readout and Verify Program	General Electric	-4	4-8
11. GAEC Hybrid/Physical Simulation	GAEC	-6	4-7

The engineering guided flight simulation is sometimes known as the Ideal Flight Simulation since accurate mathematical expressions are used in the navigation and guidance computations.

6.2 MIT INTERPRETIVE COMPUTER SIMULATION

This is a combination of an interpretive computer simulation and the vehicle dynamics, environmental, and equipment models which can be used to provide guided flight simulations of selected phases of the mission. The interpretive computer portion is capable of simulating in a bit-by-bit fashion the operation of the AGC and LGC. The vehicle dynamics model should contain the rotational dynamics as well as the translational dynamics.

This simulation should be designed to operate in several modes. The first is an open loop or "bench test" mode in which the interpretive computer portion is provided nominal mission inputs in a form acceptable to the AGC and LGC. This mode is used during the flight programming and program checkout phases of software preparation.

The second mode is a guided flight simulation mode in which the loop is closed around the vehicle dynamics and environment. This version is used in the targeting verification and software qualification test procedures.

A third mode would take advantage of the relatively detailed vehicle dynamics and environmental portions of the simulation but would replace the interpretive computer routine with a computer hardware oriented set of flight equations programmed in scientific computer (MAC) language. This simulation would form the basis of the targeting program. This version should closely approximate the results obtained with the second version described above, but would require less real run time by a factor of about five. This version would be used to obtain flight constants by an iterative technique.

In the qualification test simulation of the RCS or coast phases of long duration, it would be reasonable to obtain attitude control system limit cycle characteristics only over representative periods during this phase and also for the periods before and after a mission sequencing event.

6.3 MSC MICROSIMULATION

As currently planned, the microsimulation being developed at MSC is a very generalized and detailed simulation program designed for complete flight proof testing of the flight software. It is being designed, programmed and checked out in modular form with three main sections. The first section consists of an interpretive computer section, which is capable of an exact duplication of the AGC/LGC computations. The second section, called an environment section, contains detailed mathematical models of the vehicle, propulsion, and attitude control systems, flight hardware, and gravitational potential and atmospheres. The last major section is called the communicator, which performs the interface function between the first two sections. The communicator section also contains the priority interrupt logic and can be used to input telemetry link and later crew input functions.

This simulation is being designed to contain detailed models of sloshing and bending motion and extremely detailed equipment models.

Because of the ambitious nature of this program, it is suggested that advantage be taken of the modular construction of the program by checking out those portions of the program which are needed on a priority basis first. This may also apply to the qualification tests, particularly if the simulation computer memory limitations make piece-wise testing unavoidable. It is recommended that two modes of operation be provided on a first priority. The first is a high frequency, rotational and translational dynamics version which will be used to perform the trade offs between the guidance steering and control system gains. This version will not require the sloshing and bending modes to provide the basic trade offs, but should include the essential features of the steering equations and control systems.

The second high priority mode of operation is a combination interpretive computer guided flight simulation used to support the software qualification testing. The fullest advantage should be taken of any studies performed by NAA/S&ID and GAEC to simplify modeling.

On a lower priority than the first two, the sloshing and bending modes could be incorporated with higher frequency models of the control system and hardware dynamics. This high frequency testing must be performed with extremely small digital integration time steps and, hence, should be employed in short, representative durations. Primary analysis of these higher frequency modes should be performed with hybrid (analog/digital) simulations. The ability to perform open loop or bench test studies on the flight software with special emphasis on diagnostic print-out and plot routines and the exercising of all possible interrupt signals from the data link would also be of lower priority, since this would normally be performed by MIT as part of the program checkout process. The open loop response engineering tests discussed in the MIT interpretive description need to be repeated only if some questionable results are found.

For the qualification tests, the comments made in the MIT micro-simulation section concerning simulation during the RCS or coast phases also apply here.

6.4 HYBRID (ANALOG/DIGITAL) 6-DEGREE-OF-FREEDOM SIMULATION

The primary purposes of this MIT hybrid (analog/digital) computer simulation are the design and verification of digital control system equations and selection of constants, the verification of adequate control system stability margins and performance including high frequency modes, the verification of the guidance/control interface, the qualification of simplified digital models, and verification of proper phasing of attitude signals and steering commands. To accomplish the majority of these purposes, inclusion of simple models of the spacecraft control system is sufficient, the exception being high frequency dynamic control system studies. This simulation can be useful in performing realistic parametric re-entry studies.

For the verification of control system gains, limits, and digital filters, it is recommended that open loop response tests of the programmed equations be made to insure that the precise constants selected have been included. Transient responses of the equations as a result of input initial conditions and functions can be precisely determined, confirming the selected gains and filters, and saturating signals will verify that the limits are proper. These results are far more accurate and predictable than those for the closed loop simulations which are either subject to hardware tolerances or to simulated hardware nonlinearities. The open loop response tests can be viewed as being an engineering verification of the programming of control equations and constants.

A man-in-the-loop or human operator simulation will be desirable in the verification of manual control logic equations and, to a greater degree, in the verification of programs associated with the optical subsystems. An engineer with an operations handbook can monitor and verify the manual procedures. It may be necessary to include part of the actual G&N hardware in order to conduct the man-in-the-loop portion of this simulation.

6.5 HYBRID/PHYSICAL SIMULATION

The primary purpose of this MIT Analog/Digital/Physical Simulation is to aid expeditiously in the verification of mechanical and electrical interfaces for the core ropes, AGC/LGC, and other spacecraft G, N and C equipment. This simulation is used to provide functional checkout of the tapes which complement the bit by bit check Raytheon has performed on the ropes. It can be developed as an extension of the MIT combined analog/digital simulation where G,N and C hardware replaces these digital models. This simulation can also be used to verify and develop equipment integration tests.

6.6 PUNCHED TAPE AND CARD DECK GENERATOR AND COMPARATOR

A program to convert either a magnetic tape or card deck of the flight program memory to punched tape or cards in a given format should be available at MIT. This program would provide the punched tape for the wiring machine input, the check tape, and the variable or destructive readout (DRO) tape input in formats applicable to each purpose. This program should also be capable of comparing one form against the other on a bit-by-bit basis to verify the tape preparation process. Once this program is established, it need not be modified unless changes in format are required.

6.7 NAA/S&ID HYBRID/PHYSICAL SIMULATION

This NAA/S&ID Combined System Dynamic Verification (COSYDYVE) simulation program incorporates, as much as possible, all of the spacecraft flight hardware of the G&N and the SCS system. It can be developed as an extension of a hybrid engineering simulation with optional capability to include each hardware item separately.

The simulations conducted by NAA/S&ID and GAEC are normally not required in the software validation program. However, because they are available, they can be used to perform the function of validating the simulation models and equipment interfaces.

The value of these simulations is greatest during the initial phase of the program and as support for vehicle integration tests and subsequent hardware changes. They can also be of considerable aid in post flight evaluation and reconstruction of hardware malfunctions. This simulation includes an AGC and a core rope simulator for accepting both core rope and tape inputs. The COSYDYVE simulation can also be useful in the testing of contingency flight configuration employing released ropes.

Use of the COSYDYVE simulation might assist significantly in formulating and conducting vehicle integration tests. Perceptive tests of problem areas and interfaces can be developed and the better understanding of test constraints and difficulties will enable construction of an effective test program. These may be sufficient reasons to supply NAA/S&ID and GAEC with a copy of the release flight rope for each mission. Any software testing included in the use of these simulations will land added confidence to the software, however.

6.8 WIRING MACHINE PROGRAM

The wiring machine program is prepared by Raytheon to operate the wire machine from a punched tape. The wiring of core ropes is a semiautomatic process in which the tape controls the thread/no thread decisions for each of the sense lines when the rope is fabricated. The input format requirements of this program are formalized in the Wiring Machine Input Requirements Document to insure that the wiring tapes received by MIT have the right format. Once this program is established and documented it need not be repeated unless the wiring machine or its input requirements change.

6.9 MEMORY READER AND CHECK TAPE COMPARATOR

The memory reader and check tape comparator program is required by Raytheon to test the correctness of the flight rope fabrication. It should be capable of comparing the memory read from a fabricated rope with that provided in a possibly different format from the check tape on a bit-by-bit level. This program would not require revisions once it has been established.

6.10 DRO TAPE READ IN, LGC/AGC READOUT AND VERIFY PROGRAM

This program is provided by General Electric as part of the ACE computer program. It should be capable of controlling the loading of the destructive readout (DRO) or temporary memory of the AGC/LGC and the subsequent readout and verification of that portion of the memory. This program is not mission dependent and will remain virtually unchanged unless computer loading and readout capabilities are modified. It will have to be revised for the change over to Block II computer, however.

6.11 GAEC HYBRID PHYSICAL/DIGITAL SIMULATION

This GAEC simulation for the LEM system is comparable in scope to the NAA/S&ID COSYDYVE simulations. The simulations includes a LGC and a core rope simulator for accepting both core rope and tape inputs with as much interfacing LEM spacecraft hardware as possible. The majority of the comments for the COSYDYVE simulations are applicable to the GAEC simulations with the exception that all of the LEM primary G&N systems will employ digital guidance and control system equations and no Block I or Block II designation is employed.

7. SOFTWARE TEST TECHNIQUES

This section describes some of the important aspects of the equation tests, the program checkout, and the qualification tests which can be used as a guide in preparing the appropriate test plans. The specific simulation runs can only be defined on a flight-by-flight basis taking into account the specific software involved and the modifications taking place from previously verified subprograms. If the software subprogram standardization recommended here is implemented, much of the testing required for later flights can be minimized by building on the backlog of previous tests. These tests can be supplemented with the special tests required for the specific mission design changes, retargeting of contingency flights, configuration changes, reorganizing of the program assemblage, or software refinement found necessary from previous flights. These test techniques discussions will therefore be general in nature.

7.1 EQUATION TESTS

The equation tests should be performed prior to release for programming and should be designed to establish that the equations have been designed in accordance with the requirements of the Equation Specification. The tests should contain, as a minimum, a series of simulation runs to evaluate the performance of the equations in the presence of nominal and non-nominal conditions and to evaluate the correctness of the logical design of the automated decisions and sequencing.

Specifically, simulation runs should be designed to evaluate the ability of the equations to reproduce the nominal mission design under nominal conditions. The limitations imposed by the approximations to the real world often found necessary in the equation design should be evaluated or the appropriate biasing techniques evaluated. The targeting requirements should be identified in terms of adjustments in constants required to reproduce varying mission conditions.

Most of the equation tests can be performed on an engineering version of a three-dimensional guided flight simulation, although often the realistic simulation of re-entry requires a six-degree-of-freedom simulation. Other portions of the equation test can be performed on specialized simulation programs such as an optical equipment simulation for lunar navigation, for example.

The ability of the equations to properly control the trajectory in the presence of an exhaustive list of 3σ or maximum values of vehicle, propulsion, environmental, and sequencing dispersions should be evaluated. This list should be established as part of the equation specification and should be current in order to realistically assess the performance of the equations with respect to the mission derived performance criteria, also established in the equation specification. Special studies should be made to determine the dependence of software performance with respect to guidance hardware sensor errors so that realistic error analysis procedures can be established. The linearity and correlation between effects of perturbations and relative sensitivites to perturbation parameters should be established to aid in the extrapolation of performance estimates under proposed mission, environmental or measurement error changes. The understanding of the equation performance gained during these tests will also permit the selection and combination of limited perturbations in the targeting verification operation. This is where an efficient but realistic and stringent test of the software performance can be made as applied to a specific mission phase once the proper constants are determined. Tests should be designed to establish the validity of the mission sequencing logic in switching between computer operating modes or responding to all alternate means of external updating including the command interrupt logic. At this point external influences are preprogrammed into the simulation tests rather than performed in a more real time sense. However, the real time data input/output requirements are evaluated with respect to format.

The equations should be exercised in all abort and contingency modes not only for the specific mission in preparation but for the operational mission conditions, if time permits. These test results can aid in the standardization of equation subprograms and minimize the testing and software modifications required for subsequent missions.

The equations should be tested for their characteristics recovery from various and dispersed initial conditions so that the same subprograms may be used in several mission phases where possible and their limits of application established.

An important aspect of the equation tests which is often overlooked is the testing of equations to destruction, so to speak. Combinations of perturbations or unrealistically large perturbations, for example 10-15 σ, should be included to gain an understanding of the weak spots and failure modes of the equations and their limits of predictable performance or characteristics of performance deterioration. This information can be used to design reasonableness tests on constants, particularly when they are subject to external modification by the uplink or the astronaut through the DSKY. This information can be used in the design of diagnostic tests and self correction logic often found useful.

The equation tests can also be used to justify simulation model simplifications where the behavior of the simulated guided trajectory can be compared using simplified control system models or RCS logic, for example. The effect of integration step size, the frequency spectrum of the vehicle, propulsion and equipment dynamic models can be evaluated. Generally, it is found that the basic behavior of the guided flight simulation is not changed significantly when more realistic details are added to the simulation models. This information also provides the basis for more efficient performance analysis and targeting computer runs in the future.

By writing an equation test plan early in the equation development phase, many of the results of the informal engineering and tradeoff tests normally performed anyway in equation development can be recorded as part of the test results,thus leaving to the end only the more formal demonstration of performance required to satisfy performance criteria.

7.2 PROGRAM CHECKOUT

The AGC and LGC flight program checkout should occur after the equation testing phase, and before qualification testing. The primary purpose of this checkout is to verify that the equations which have been coded and programmed in the flight computer language, agree with those described in the Guidance Equation Description Document and mechanized according to the specific requirements found in the document. These requirements include ranges of the values of the variables from which scaling parameters can be obtained, representative values of the constants used in the equation, for comparison with equation test results, and any special programming techniques to be employed such as special overflow protection, and constraints on computational frequencies, etc.

Program checkout consists of a series of tests which are performed on various flight equation configuration levels. This technique is used because it allows a systematic build-up of confidence in the software. In these tests, an interpretive computer simulation (ICS) is employed in an open-loop manner (i.e., "bench-testing"). A complete nominal mission profile is used as a driver in these simulation studies. This driver must contain parameters which are compatible to all of the input logic used in the simulation program, and must provide outputs compatible with the computer interface requirements.

The first series of program checkout tests involves a study of the characteristics of the programmed flight equations subroutines, and overall program assembly. This involves checking the program assembly for duplicate or unassigned locations, finding intermediate quantities in the program which are beyond the computer scaling provided, and checking for errors in the defined arithmetic operations to be performed. A check should be made to assure that the value of the constants used in the programmed equations agrees with the values specified in the equation description document.

The second series of tests involves performing limited duration open loop ICS runs to evaluate the performance of individual subroutines used in the program. This is accomplished by subjecting these subroutines to input data which varies over the expected dynamic range of these variables. Toward this end, three different values could be used for each input constant. They would consist of an expected minimum value, intermediate value, and the maximum value. The performance of each subroutine would be analyzed to assure that it functions properly throughout the anticipated range of computational variables. Branching logic and abortive logic will be examined to verify its operational performance.

The third series of tests is performed on the total guidance program which contains all the required subroutines and interfacing logic. The input stimulus for these tests is the nominal mission profile driver. The results are then compared with the anticipated results obtained in the engineering tests. Included in these simulation runs should be tests which can be used to establish the time-sequencihg limitations of data uplink quantities, ground program computation time intervals, and astronaut-flight computer reply times. This involves establishing both the time increment requirements needed to perform these functions and the interface timing requirements. This information can be used in determining the expected running times for all of the flight and ground software. This is particularly important when verifying the priority interrupt logic used in branching to any alternate guidance subroutines in the program. These simulation runs should also include tests which exercise the upper and lower bounds on the constants used in the program. This provides a check on the integrity of the constants in the program and a means of determining the validity of the scaling employed. All mode switching command logic should be thoroughly tested to assure that the priorities and computations used are commensurate with requirements stated in the Guidance Equation Description Document.

The results of the program checkout tests are used to indicate any discrepancies which might have occurred between the finalized guidance equations from the engineering simulation studies and the programmed flight computer equations. They are particularly beneficial for indicating where guidance modes should be modified, or revised programming techniques employed to correct any marginal guidance logic in the AGC and LGC programs. At the completion of program checkout, the equations document should be revised to faithfully describe the programmed equations and constants.

7.3 QUALIFICATION (SYSTEM SIMULATION) TESTS

The qualification or system simulation tests are performed after program checkout has been completed, and is designed to demonstrate the overall performance of the guidance and navigation system when operating as part of the astronaut/vehicle combination. Its most distinguishing feature is the use of the flight software, as implemented in the AGC or LGC, by means of closed loop interpretive or micro-simulations. The tests should be sufficient to demonstrate the behavior of the equations as implemented in the guidance computer with the scaling specified in the Guidance Equations Description Document (and subsequently used in guidance computer programming). The performance of the G & N System is evaluated under nominal and abnormal vehicle, environmental and input data performance conditions including marginal limits which cannot be realistically determined during equation testing. Since the combination interpretive computer simulation/flight simulation is relatively expensive to run, as much software testing as possible should be done during the Equations Test. However, the realistic testing of computer scaling, computational timing, command interrupt logic, and dynamic compatibility of the computer input and output with their interfaces, can only be realistically evaluated with a bit-by-bit or at least a word-by-word or instruction-by-instruction simulation of the software.

The Qualification Tests include specific tests for computer scaling under expected nominal and non-nominal conditions during portions of all phases and modes of computer operation. The timing and sequencing used in the guidance program, which are not covered in the program checkout, is verified. The computer logic involving mode switching and command interrupt is thoroughly exercised by preplanned interrupts or by an engineer at the controls of the computer exercising mission procedures during all phases of the simulation.

The interfaces between the computer and inertial platform, the optical subsystem, the astronaut/DISKY and the spacecraft subsystems are exercised. The models used in the qualification test simulations of these interfaces are justified by separate tests or hybrid simulations often made a part of the overall qualification testing. (This justification of the simulation model is the main contribution of the physical/analog/digital simulations of NAA/S&ID and GAEC to the software validation effort).

The combined performance of the G&N software and hardware is verified during qualification testing by comparing the results with the performance criteria originally defined in the software specification. This is often of the nature of a proof test in that earlier tests have established their performance separately.

The qualification tests can be performed with targeting constants derived from a representative mission, not necessarily the specific mission to be flown. But where possible, enough of the mission variations should be included to demonstrate the manner of incorporating launch time dependent constants, and the compatibility with the targeting technique. To aid in the standardization of the software subprograms, a range of anticipated operational missions should be included where time permits. The targeting verification will insure that the specific mission requirements will be met.

The qualification tests should include at least one complete mission simulation with the ICS/FS. Since operational missions may last many days, and the ICS/FS simulation generally runs between 5 to 10 times real time, the simulation program should be designed to run in segments for periodic review and analysis. This can be done by providing the capability to start and stop the simulation at any place and store all of the intermediate data for continuation at a later time. Every effort should be made to operate the ICS/FS simulation as efficiently as possible as the qualification tests can become a significant part of the cost of the software validation program. This can only be done by investing much effort into simplifying the simulation models to the greatest extent possible without significantly affecting the performance of the results. It is very doubtful that simulation models to the level of duplicating sloshing and bending modes will be necessary for extended simulation runs, for example. There does not appear to be a need at this point to simultaneously simulate several or all of the airborne guidance computers at the same time to validate the software. This may be a problem in the real time ground program, but the determination of this requirement is beyond the scope of this study.

A significant part of the qualification test involves the testing of the software to destruction. To really exercise the scaling, alarm discretes, command interrupt, abort and diagnostic logic, reasonableness tests on DSKY and uplink inputs, etc., unreasonably high values of perturbations and purposeful attempts to confuse the program must be simulated. The limits of its predictable behavior and the failure modes must be determined to really gain confidence in the software.

The above discussion provides some of the reasons why the software should be standardized wherever possible. Once the software is proven satisfactory, unnecessary changes should be avoided and that strict fidelity of the description documentation should be maintained. If this is implemented, the qualification testing can be limited to selected tests of software behavior in later flights, and special testing designed to verify the modifications found necessary.

8. TARGETING AND VERIFICATION PROGRAM

This section contains a discussion of the Targeting and Verification Program which is operated before each flight. The program is used to determine the specific mission or flight dependent constants and ensure the satisfaction of all mission and target requirements. The launch time dependent constants and check sums and detailed of the launch window are also generated.

It is recommended that this program be somewhat automated because of the number of similar missions and high launch rate. If changes are required in the permanent memory of the computer, then a revised memory deck is also provided with the verified software package.

Basically the targeting and verification process consists of the 1) final mission trajectory design, 2) the specific mission dependent constants generation, 3) a limited verification of performance, and 4) verification of compatibility of the guidance constants and program with mission objectives.

Targeting verification should be done independently by two different groups. The verification portion of this process is intended to verify specific mission oriented flight constants as applied to a specific mission so that the particular software objectives are met. By contrast, qualification testing is a general test of the software program and may not be oriented to a specific mission. Targeting verification can be considered to be a specialized scaled down qualification test. The flight dependent constants developed as a result of the targeting effort are then fabricated into one replaceable computer memory module.

Depending on the extent of pre-targeting mission design, the trajectory shaping routine may be quite involved. The process consists of iteration between powered and free flights using the "hardware oriented" version of the software to obtain a series of reference trajectories across the launch window which satisfy all the constraints and mission objectives. These trajectories define the beginning and end of the windows as well as all of the abort modes and constraints. Checks are built into the iterations to guarantee the satisfaction of all constraints throughout.

The next step is to generate launch time and other specific mission dependent constants in a form consistent with the software format and verify that they perform as expected by guided simulations. During this process, constants are scaled and checked to insure compatibility with the rest of the software. A selected set of extreme perturbations are then simulated to test the mission effects of known weaknesses of the software.

The next step is to simulate selected reference cases using the ICS/FS with the flight software to verify performance and compatibility with mission goals, and to determine that the results are in agreement with the "hardware oriented" version of the simulation used to generate guidance constants. The final step is to convert the software into a form used to prepare the flight constant module and the DRO tape if applicable.

In the case of contingency flights were the mission objectives remain unchanged, the only preparation for these flights is retargeting and verification to provide a replacement for a mission dependent computer memory module.

Since the Targeting and Verification Program tends to be a complicated sequence of specialized computer rountines, it should be controlled and validated just as the flight software itself. The basic steps in the Targeting and Verification Process are shown schematically in Figure 8-1.

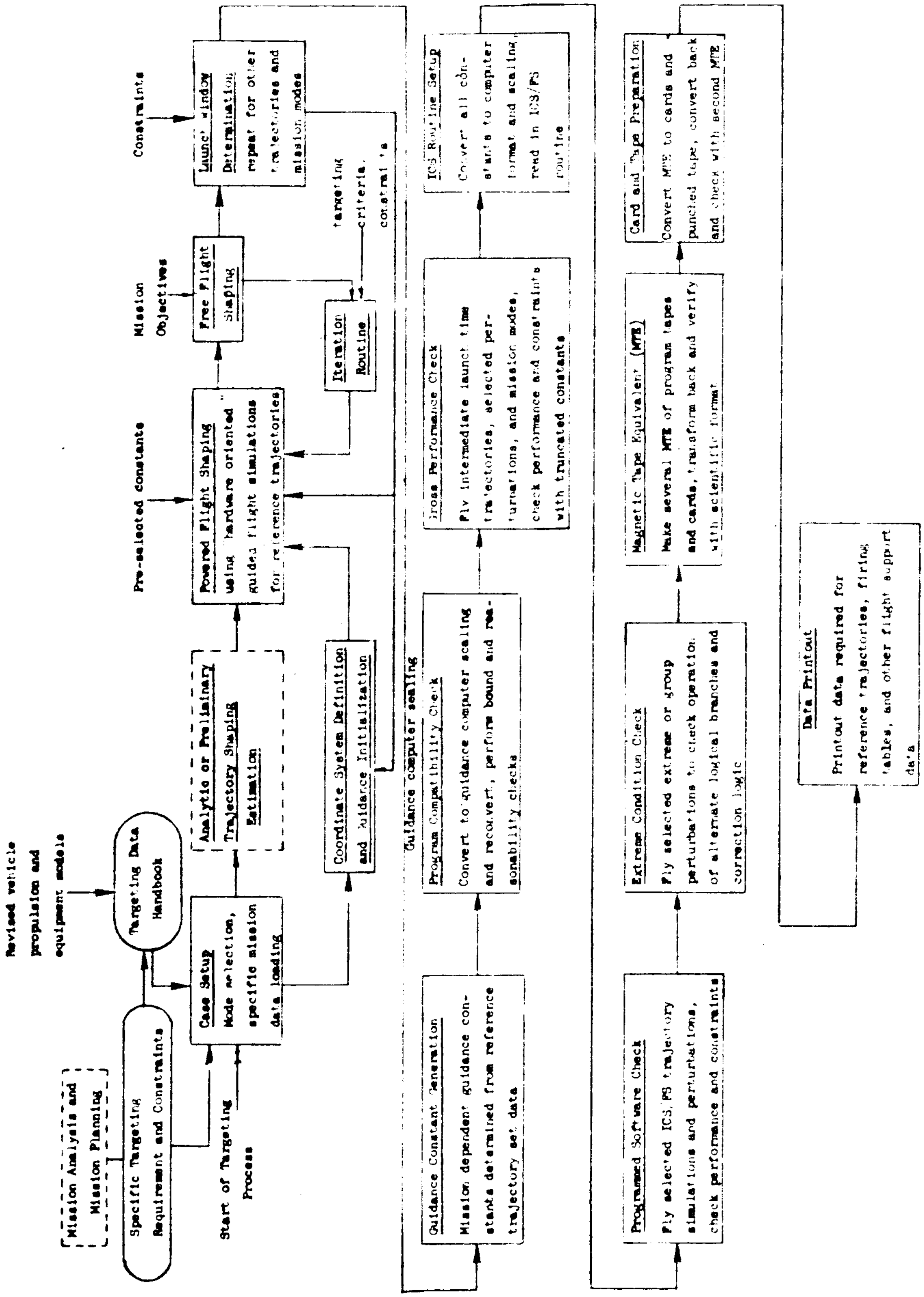


Figure 8-1. Targeting and Verification Process

9. CONCLUDING REMARKS

The application of a software validation plan to the overall Apollo program has led to several conclusions which should be summarized here for empahsis. They are as follows:

a) Some form of software standardization is essential for the success of the Apollo flight software development and validation. Since the flight schedule demands short lead time, and the mission requirements during sequential flight tests are quite diverse, the significant validation effort requirements can lead to unmanageable problems unless some standardization of software in the form of subroutines, subprogram assemblies, or even rope modules if possible is implemented wherever possible.

b) The above standardization can only be achieved with early efforts to define and develop complete Apollo operational flight software. This implies a preliminary effort to define a complete software specification from mission planning of the lunar landing and return mission. Complete equations should be developed considering the computer timing and memory capacity constraints, etc. The testing required on a flight-by-flight basis should be supplemented with tests relating to the operational mission so that the standardized software subprograms can be incorporated into the earliest flight possible resulting in flight proven software backed up by an accumulation of applicable validation testing.

c) High confidence in flight proven software requires effective software control through frequent and meaningful design reviews. In addition to reviewing the adequacy of the software design, the review board should act as a quality control board and most importantly a configuration control board. The frequent changes to the software which is inevitably desired from the pressures of changing mission planning should be resisted whenever possible with recourse to the MSC Apollo Change Control Board as necessary.

d) The Apollo program plan requires sufficient software development and validation resources to sustain at least three simultaneous flight preparation programs. The necessary man power, simulation facilities with insured computer time available, and design review board personnel who can sustain the duty cycle required must be provided to sustain the validation effort. Resources to support an adequate documentation effort should be provided as it is likely to become a significant part of the validation effort.

e) Consideration should be given to redefining the role of the large hybrid physical/digital simulation programs with respect to software validation. Since their main utility to the software validation effort is to justify the hardware simulation models used, they may not require a complete set of flight ropes in support of every test flight but only when significant changes have occurred in the hardware design or interface definitions.

f) Independent qualification testing and targeting verification has been found to be useful and should be provided for in the plans of each flight.

g) A formal targeting and verification procedure should be developed and modified as necessary to conform to the mission and software, so as to confidently support the short turn around time built into the program plan.

REFERENCES

1. R. Braslau, "Guidance Software Validation Philosophy" TRW/STL Report No. 3503-6001-TU000 (NAS 9-2938) 5 April 1965.

2. Interim Report No. 2, "Recommended Apollo Flight 202 Software Validation Plan", TRW/STL Report No. 3730-6001-TU000 26 April 1965.

3. "Apollo Spacecraft Master Test Plan" (a) (First Issue), NASA/MSC Document No. TIP 545-0P04-5003 April 15, 1965.