Development of an AEGIS Maintenance Methodology

Task 2: Models of the AEGIS Maintenance Process

By Benjamin J. Keller, Richard E. Nance and James D. Arthur

TR 90-43

SRC-90-006*

DEVELOPMENT OF AN AEGIS MAINTENANCE METHODOLOGY

TASK 2: MODELS OF THE AEGIS MAINTENANCE PROCESS

Interim Status Report

Benjamin J. Keller
Richard E. Nance
James D. Arthur

Systems Research Center
and
Department of Computer Science

Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

1 August 1990

* Cross referenced with TR CS-90-43, Department of Computer Science, Virginia Tech.

This report was supported by the Naval Surface Warfare Center under contract #N60921-83-G-A165 B048.

ABSTRACT

Covering the Task 2 effort in the development of an AEGIS software maintenance methodology, the derivation of requirements from maintence principles is summarized. Three models of the maintenance process provide the basis for identifying points of application. The mid-level (aggregate) model is the major focus of the methodology definition. Recommendations address the potential uses of different modeling perspectives, better balance between product and process assessment, use of metrics, and the selection of maintenance tools.

CR Categories and Subject Descriptors: D.2.1 [Software Engineering]: Requirements/Specifications; D.2.7 [Software Engineering]: Distribution and Maintenance.

General Terms: Documentation, Management.

Additional Key Words and Phrases: Methodologies, reverse engineering, model, maintenance.

Toward an Improved AEGIS Maintenance Methodology

1. Introduction

This interim status report is the second in the project entitled, *Development of an AEGIS Maintenance Methodology* (N60921-83-G-A165-B048). The project has been divided into three tasks:

Task 1: Definition of Maintenance Methodology Requirements

Task 2: Development of AEGIS Maintenance Process Models

Task 3: Specification of the AEGIS Maintenance Methodology

This report covers the period 15 December 1989 to 15 July 1990. Briefing of the results contained in this report to NAVSWC sponsors took place on 18 July 1990.

1.1 Background

The second task, utilizing the software maintenance principles and requirements from Task 1, employs the process models under development at the Naval Surface Warfare Center (NAVSWC) by government personnel and SAS Consultants. The software maintenance principles and the generic requirements derived from them, are to be used in the development of a methodology in the classical top-down definition. However, the derivation of abstract models of *maintenance process intent*, from the operational models currently being produced, is a means of assuring that the particular domain-specific requirements for embedded system software are preserved in a hard-real-time application (see [FAUS89]). Factors related to the AEGIS application domain, such as system architecture, programming language, and organizational structure, are included from the onset in the attempt to maintain relevancy and realism.

1.2 Statement of Work for Task 2

The statement of work for Task 2, prescribing the development of the AEGIS maintenance process models, includes the activities of code modification, documentation updating, training, inter-element communications, configuration management, unit and integration testing, and software quality assurance as falling within the model. The modeling activity is cooperative in nature, with the assistance of a panel of senior software personnel representing each element of the AEGIS combat system providing in-depth process definition.

Actually, the development of the current process model has been delayed beyond the original projections, and the VT/SRC departure is from an interim representation and not the final. This delay, although not proving damaging to the derivation of subsequent models, has forced the conclusion of Task 2 to exceed the original schedule by approximately six weeks. Nevertheless, the interaction permitted between VT/SRC, NAVSWC, and the SAS Consultants is appearing to produce a current process model in which greater confidence is warranted.

In Section 2 of this report the current process model is presented. This model is reconsidered from the perspectives of process- and product-centered activities, shown in Section 3. The fourth section outlines the application of requirements to the model, and the fifth presents recommendations and directions appropriate for Task 3.

2. Models of the AEGIS Maintenance Process

The three models of the AEGIS maintenance process, constituting the subsections which follow, are constructed with different objectives in mind. The current model, also called the detailed model, attempts to represent a process as it currently is performed and understood by all those involved. This low-level model is a representation of "what is," and is necessary for the evaluation of alternatives to answer the question "what could or should be?"

The high-level process relationship model, described in Subsection 2.2 has the objective of describing the inter-process communications. The mid-level or aggregate model in Subsection 2.3 is derived from the detailed model as an expression of intent. As such, the model is intended to represent what is being accomplished in the detailed representation but excluding the more precise characterization of *how* the intent is achieved currently.

2.1 The (Detailed) Model of the Current Process

The working (low-level) model developed by NAVSWC and SAS Consultants divides the AEGIS Maintenance Process into nine subordinate processes:

(1) Establish Program Library (100)

(2) Perform Detailed Design and Code (200)

(3) Element Test (300)

(4) System Integration and Test (400)

(5) Computer Program Specification Change (500)

(6) Develop Delivery Package (600)

(7) Delivery of Baseline (700)

(8) Develop Update (800)

(9) Computer Program Change Request (900)

Each process is also identified by the number of the beginning activity in each diagram. The dates for each diagram, the version used in this work, and the current version, are shown in Table 1.

The effort on Task 2 is constrained to some extent by the necessity for "fixing" on a working version of the diagram. These diagrams have been superseded by more recent versions which may more accurately represent the current process. However, the process under scrutiny is dynamic, and the "current" process model can never be a precise, correct

Table 1. Issue Dates of Original and Supplied Versions of the Process Diagrams

Major Activity	Working Version Issue Date	Current Version Issue Date
Establish Program Library (100)	14 March 90	27 March 90
Perform Detailed Design and Code (200)	16 November 89	12 July 90
Element Test (300)	14 March 90	27 March 90
System Integration and Test (400)	14 March 90	12 July 90
Computer Program Specification Change (500)	1 January 90	2 July 90
Develop Delivery Package (600)	16 November 89	5 July 90
Delivery of Baseline (700)	16 November 89	5 July 90
Develop Update (800)	14 March 90	2 July 90
Computer Program Change Request (900)	27 March 90	2 July 90

representation. Our assumption is that precision is unnecessary in deriving a representation of intent (Subsection 2.3) or extracting the inter-process relationships (Subsection 2.2).

2.2 The (High Level) Process Relationship Model

The inter-process flow relationships among the major activities are shown in the high-level model. The diagram (Figure 1) shows that entry to the maintenance process can occur at only one of three points. The first is the receipt of a baseline shipment in Process 100. A shipment from General Electric is received, audited, and distributed to the elements. The process then terminates since no further activities are needed to process the shipment.

The second entry point leads to changes in the program specifications (Process 500). Thus, maintenance activities follow a sequence of design, coding, testing and delivery. If either element test or delivery fails, the process returns to the design and coding activities. On successful delivery the process terminates.

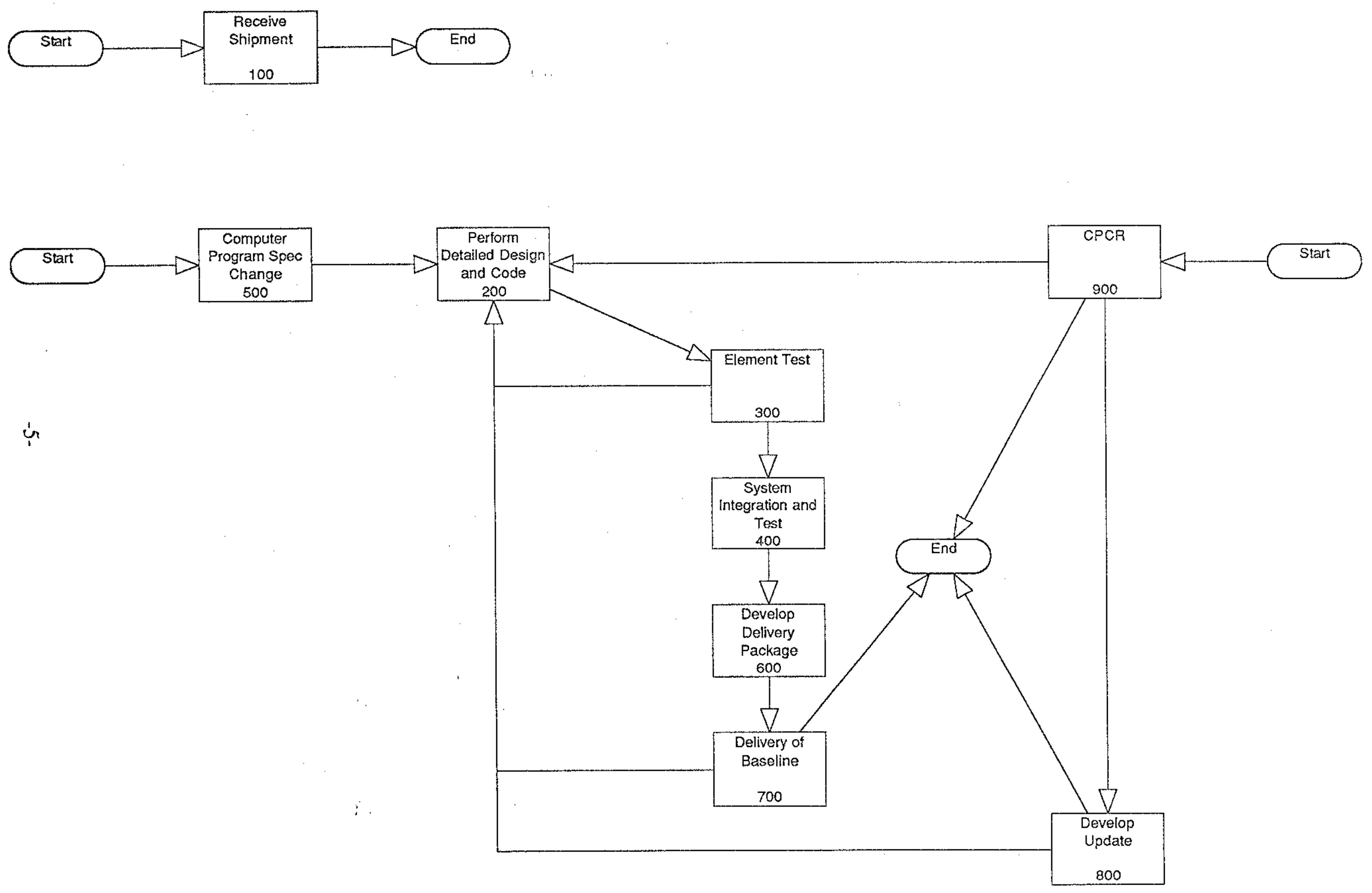

Start
Receive Shipment 100
End
Start
Computer Program Spec Change 500
Perform Detailed Design and Code 200
CPCR 900
Start
Element Test 300
System Integration and Test 400
Develop Delivery Package 600
Delivery of Baseline 700
End
Develop Update 800

Figure 1. Process Relationship Model

The third entry point is through the analysis of, and plan of action regarding, Computer Program Change Requests (CPCR's). CPCR's specify changes in response to identified problems in the system. They are handled by either going through the design to delivery sequence or by a rapid update process which generally creates and applies a patch to the system and finishes with delivery. If an update cannot be made by patching, then the normal design to delivery sequence must be followed.

Although the procedural relationships among the processes are shown by this model, the model ignores description of process functionality. A data flow diagram (DFD) model might better portray the functional roles. The DFD in Figure 2 shows the relationship of the activities to the resulting products. Each major activity is related to some product. This DFD is based on the detailed diagrams and the associated descriptions provided by NAVSWC and also from discussions with NAVSWC personnel, and is provided to illustrate the potential benefit of alternative modeling forms.

In Figure 2, the entry points are shown as being triggered by inputs from both inside and outside the process. First the delivery of a baseline initiates the receipt of shipment activity (which is connected to the rest of the process by generation of CPCR's for discovered problems). Secondly, the specification change activity acts on approved specification changes (SC's) from NAVSEA and GESD. Thirdly, the CPCR activity is fed by the collection of CPCR's generated by other activities in the process. Completion and further study of the DFD model may reveal relationships among activities and outputs that offer added insight for improvements.

2.3 The (Mid-Level) Aggregate Model

The aggregate model of the AEGIS maintenance process is formed with the goal of identifying the methodological intent of the activities. This model is derived by aggregating process activities to describe clearly identifiable functions. The aggregation is shown in the

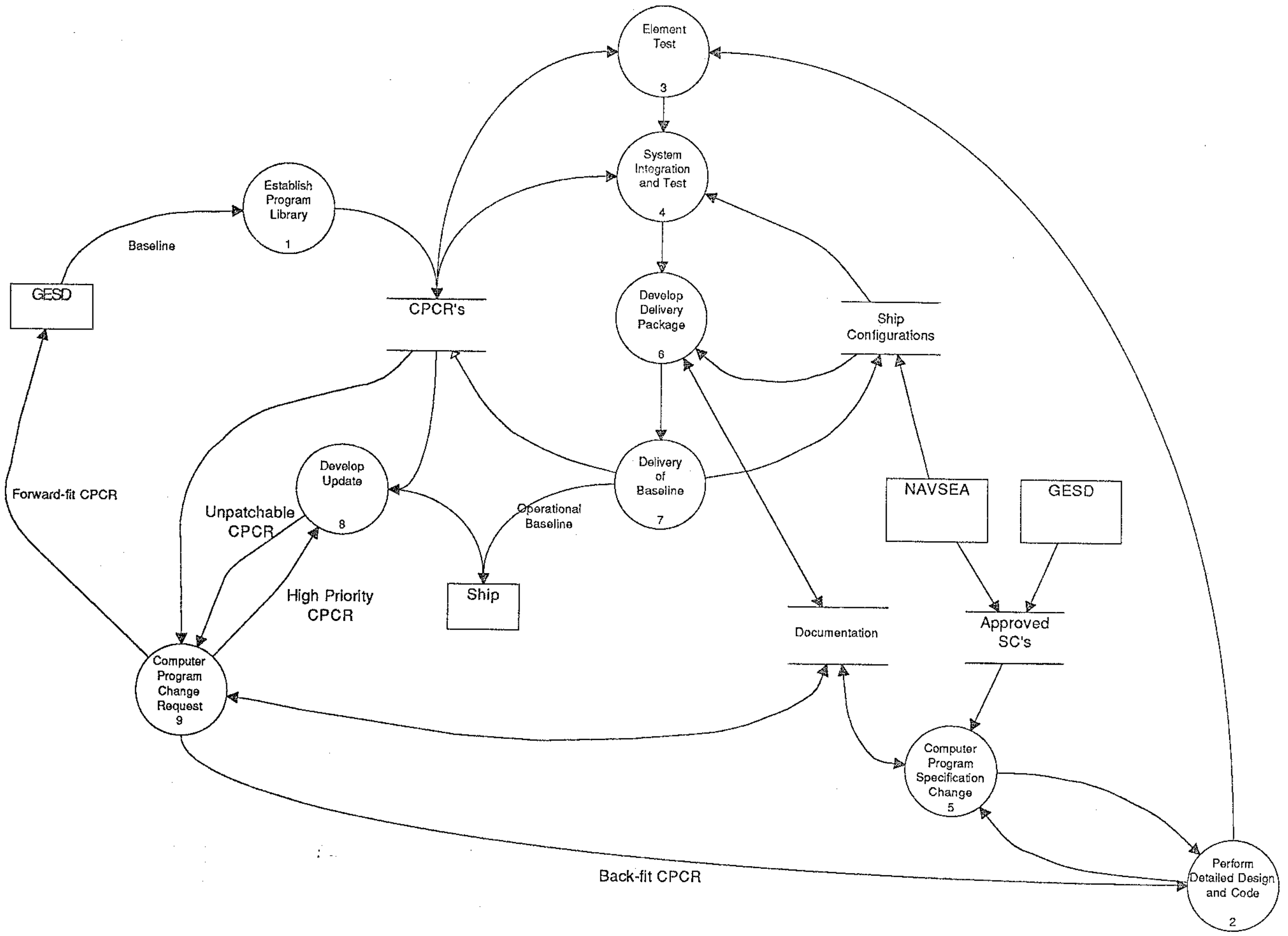


Figure 2. Data Flow Model of Process Relationships

diagrams provided in Figures 3-11. The structure of each diagram is basically the same as for the current detailed model. To preserve correspondence between the two models, the numbers of combined activities are shown on the lower left of each diagram symbol. On the lower right of each symbol are identities of the performing organization(s) for all of the activities aggregated within that symbol.

2.3.1 Receive Shipment

The first major activity is Receive Shipment (Establish Program Library) or Process 100. This process is intended to: (1) receive the shipment of a baseline from General Elective System Development (GESD), (2) confirm that the baseline conforms with expectations, and (3) distribute the baseline to the elements. Process activity is depicted in Figure 3.

The major impact on maintenance by Process 100 is in terms of preparation. In this activity the documentation and code are placed in libraries where the performers of later activities can access them. Additionally, software quality assurance (SQA or QA) must analyze the delivery to establish a standard for audits of the maintenance performed on this baseline.

Another important point in that Process 100 has no predecessor processes. The influence of this activity is through the formation of the libraries and generation of CPCR's against the new baseline. Neither influence is shown as such by the current process model.

2.3.2 Perform Detailed Design and Code

The second major activity is Perform Detailed Design and Code (activity 200). This process, as shown in Figure 4, begins with either a CPCR or SC (which generally results in one or more CPCR's). The change indicated by the necessitating CPCR or SC is first identified and analyzed for its acceptability. On approval, a detailed design is developed, coded and unit tests performed. (Note that these tests seem to be mostly against the CPCR,

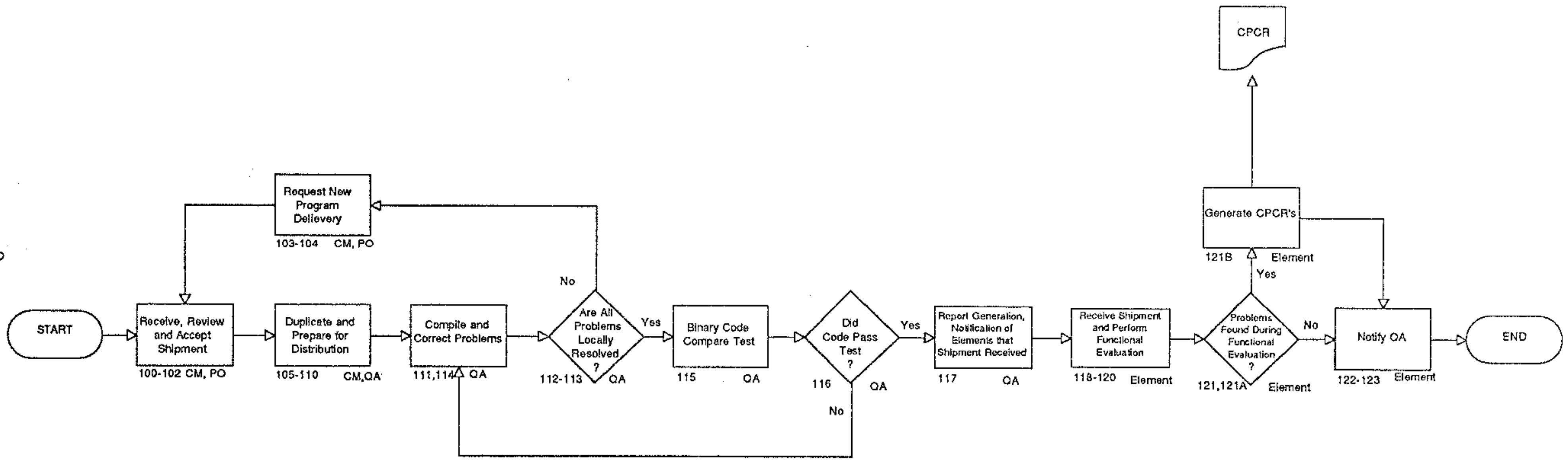

START
Receive, Review and Accept Shipment
100-102 CM, PO
Request New Program Delievery
103-104 CM, PO
Duplicate and Prepare for Distribution
105-110 CM,QA
Compile and Correct Problems
111,114 QA
Are All Problems Locally Resolved ?
112-113 QA
No
Yes
Binary Code Compare Test
115 QA
Did Code Pass Test ?
116 QA
No
Yes
Report Generation, Notification of Elements that Shipment Received
117 QA
Receive Shipment and Perform Functional Evaluation
118-120 Element
Problems Found During Functional Evaluation ?
121,121A Element
Yes
No
Generate CPCR's
121B Element
CPCR
Notify QA
122-123 Element
END

Figure 3. Receive Shipment Process (#100)

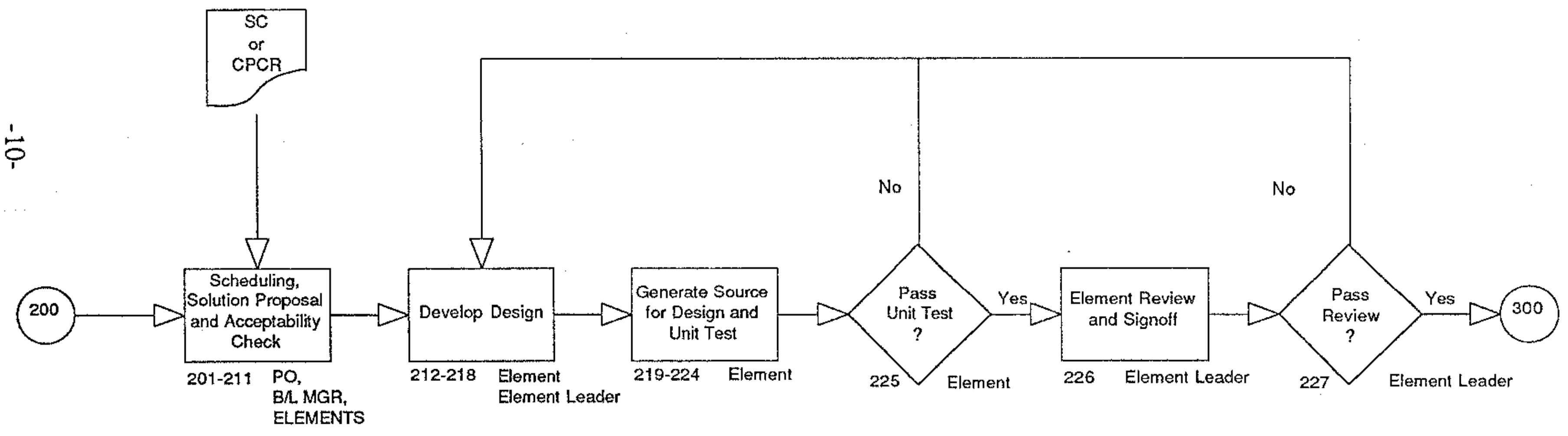

SC or CPCR
200
Scheduling, Solution Proposal and Acceptability Check
201-211 PO, B/L MGR, ELEMENTS
Develop Design
212-218 Element Element Leader
Generate Source for Design and Unit Test
219-224 Element
Pass Unit Test ?
225 Element
No
Yes
Element Review and Signoff
226 Element Leader
Pass Review ?
227 Element Leader
No
Yes
300

Figure 4. Perform Detailed Design and Code Process (#200)

rather than a unit test in the standard sense which would be against a specification.)

Most of Process 200 is performed at the element level, although Program Office and Baseline Manager involvement occurs initially. The final check is performed by the element leader, who decides if the change has been implemented and the test results justify initiation of the element test.

2.3.3 Element Test

The Element Test, performed in Process 300, is shown in Figure 5. This testing is performed by the element following implementation of fixes to several CPCR's (those which make up the scheduled changes in the baseline). Both unit and integration testing are done, and the conformance to specifications of intra- and inter-element functions are examined. The dependence of the system on combat system hardware, and its lack of availability, requires that simulations of the hardware be used in testing at this level.

Following a successful element test, a Multiple Element Integration Test (MEIT) is performed. This is the first point in testing where the elements are tested with all inter-element communications represented. The testing of interfaces is the prior concern. The results of these test have implications for the continuation to system test. In addition to the generation of CPCR's for uncovered problems, an audit of the test is performed to determine preparedness for system test. Such audits should prevent incorrect or poorly-tested elements from proceeding to system test.

2.3.4 System Test

System test (Process 400) is initiated by creating the load files for testing (Figure 6). At this point patches must be reapplied and tested before beginning system test. System test is intended to exercise the system within its final operating environment to show conformance with specifications. Following this test a Delivery Readiness Review is held by the Program Office to determine if the system should be delivered. Involved in this

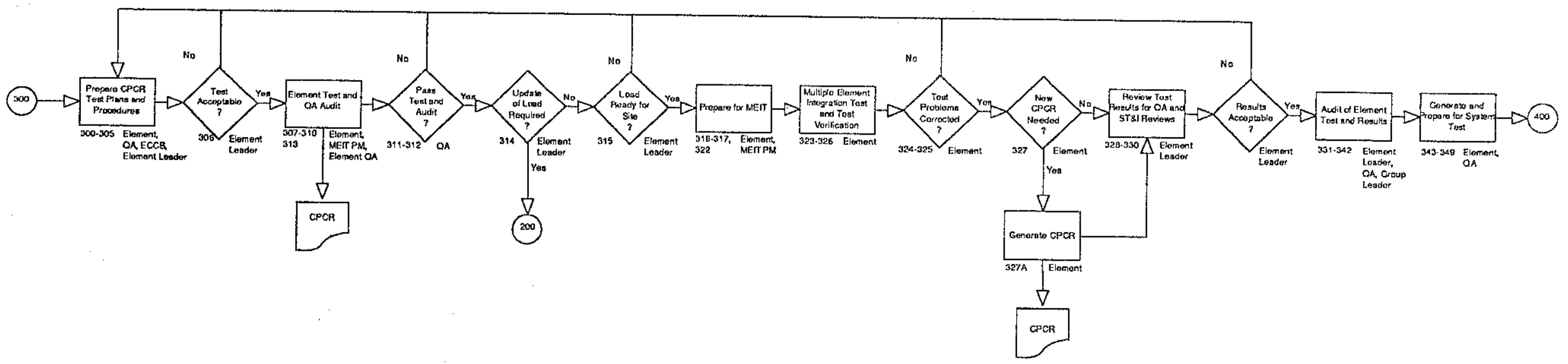

300
Prepare CPCR Test Plans and Procedures
300-305 Element, QA, ECCB, Element Leader
Test Acceptable ?
306 Element Leader
No
Yes
Element Test and QA Audit
307-310 313 Element, MEIT PM, Element QA
CPCR
Pass Test and Audit ?
311-312 QA
No
Yes
Update of Load Required ?
314 Element Leader
Yes
200
No
Load Ready for Site ?
315 Element Leader
No
Yes
Prepare for MEIT
316-317, 322 Element, MEIT PM
Multiple Element Integration Test and Test Verification
323-326 Element
Test Problems Corrected ?
324-325 Element
No
Yes
New CPCR Needed ?
327 Element
Yes
Generate CPCR
327A Element
CPCR
No
Review Test Results for QA and ST&I Reviews
328-330 Element Leader
Results Acceptable ?
Element Leader
No
Yes
Audit of Element Test and Results
331-342 Element Leader, QA, Group Leader
Generate and Prepare for System Test
343-349 Element, QA
400

Figure 5. Element Test Process (#300)

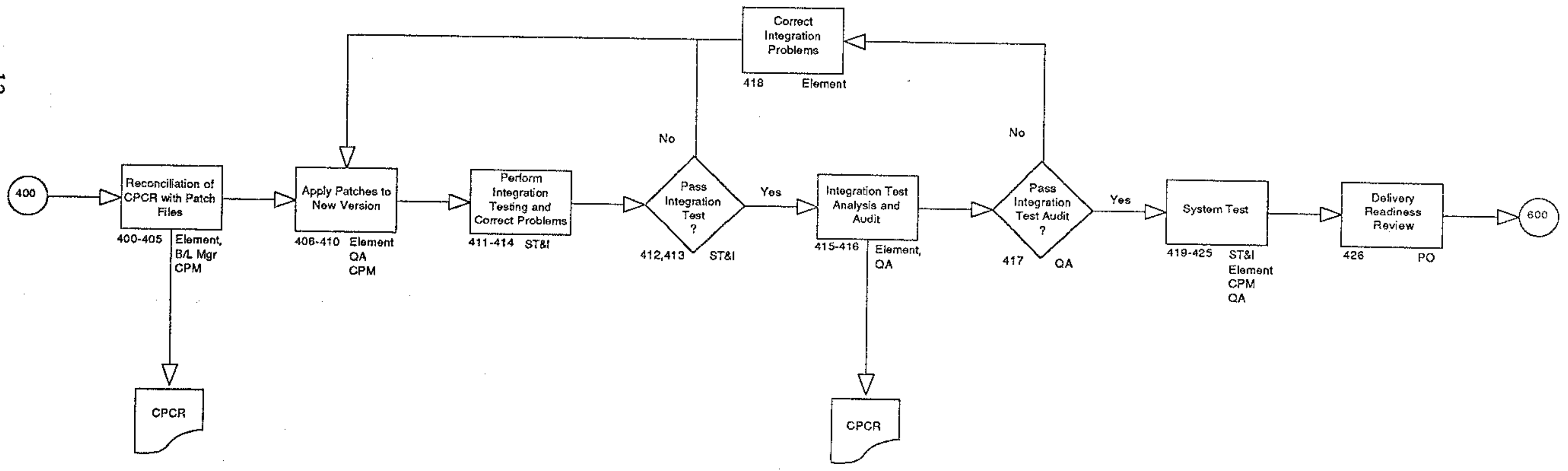

400
Reconciliation of CPCR with Patch Files
400-405
Element, B/L Mgr CPM
CPCR
Apply Patches to New Version
406-410
Element QA CPM
Perform Integration Testing and Correct Problems
411-414
ST&I
Pass Integration Test ?
412,413
ST&I
No
Yes
Correct Integration Problems
418
Element
Integration Test Analysis and Audit
415-416
Element, QA
CPCR
Pass Integration Test Audit ?
417
QA
No
Yes
System Test
419-425
ST&I Element CPM QA
Delivery Readiness Review
426
PO
600

Figure 6. System Integration and Test Process (#400)

review are all parties having knowledge of both the quality of the system (SQA and ST&I) and its preparedness (i.e., Documentation Management and the elements).

Concern has been raised by some at NAVSWC that the testing process shown in the diagrams for activities 300 and 400 does not represent actual practice. Clearly the testing process is that most affected by the real-time embedded nature of AEGIS, and careful attention to testing is essential.

2.3.5 Specification Change

Computer Program Specification Change, Process 500 is diagramed in Figure 7. Within this process, maintenance can be initiated since the goal is to respond to specification changes coming from an informal activity beyond the organizational boundaries, which include NAVSEA, GESD, and NAVSWC. In Process 500, SC's are assigned to a baseline. The selected SC's are included in supporting documentation and subject to high-level and detailed design and associated reviews. On completion of the CDR and approval by NAVSEA, the SC's are then passed onto Process 200.

2.3.6 Delivery Package

Figure 8 describes Process 600, the Delivery Package Development (600). This process follows system test and a positive decision in the Delivery Readiness Review. The activities comprising this process includes the creation of load files, delivery plans, and documentation necessary for ship delivery. An audit follows, to ensure readiness for delivery.

The delivery is performed in Process 700: Delivery of Baseline, see Figure 9. Delivery requires organization of installation, crew instruction, and testing. The test at installation allows a final test of the system in its operational environment. In this test any ship configuration peculiarities must be emphasized.

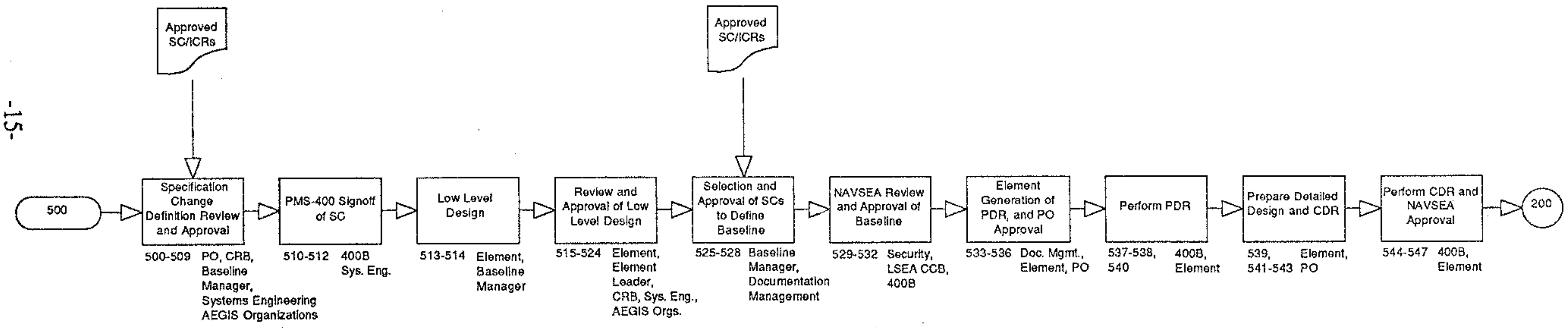

500
Approved SC/ICRs
Specification Change Definition Review and Approval
500-509 PO, CRB, Baseline Manager, Systems Engineering AEGIS Organizations
PMS-400 Signoff of SC
510-512 400B Sys. Eng.
Low Level Design
513-514 Element, Baseline Manager
Review and Approval of Low Level Design
515-524 Element, Element Leader, CRB, Sys. Eng., AEGIS Orgs.
Approved SC/ICRs
Selection and Approval of SCs to Define Baseline
525-528 Baseline Manager, Documentation Management
NAVSEA Review and Approval of Baseline
529-532 Security, LSEA CCB, 400B
Element Generation of PDR, and PO Approval
533-536 Doc. Mgmt., Element, PO
Perform PDR
537-538, 540 400B, Element
Prepare Detailed Design and CDR
539, 541-543 Element, PO
Perform CDR and NAVSEA Approval
544-547 400B, Element
200

Figure 7. Computer Program Specification Change Process (#500)

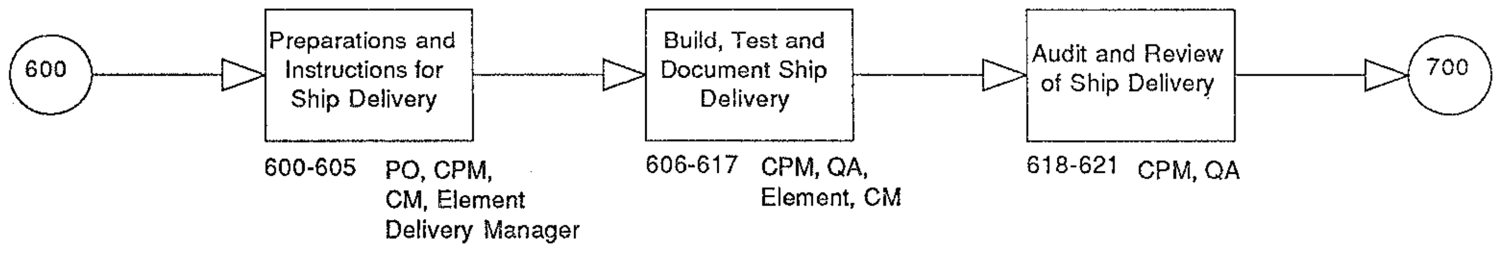

600
Preparations and Instructions for Ship Delivery
600-605 PO, CPM, CM, Element Delivery Manager
Build, Test and Document Ship Delivery
606-617 CPM, QA, Element, CM
Audit and Review of Ship Delivery
618-621 CPM, QA
700

Figure 8. Develop Delivery Package Process (#600)

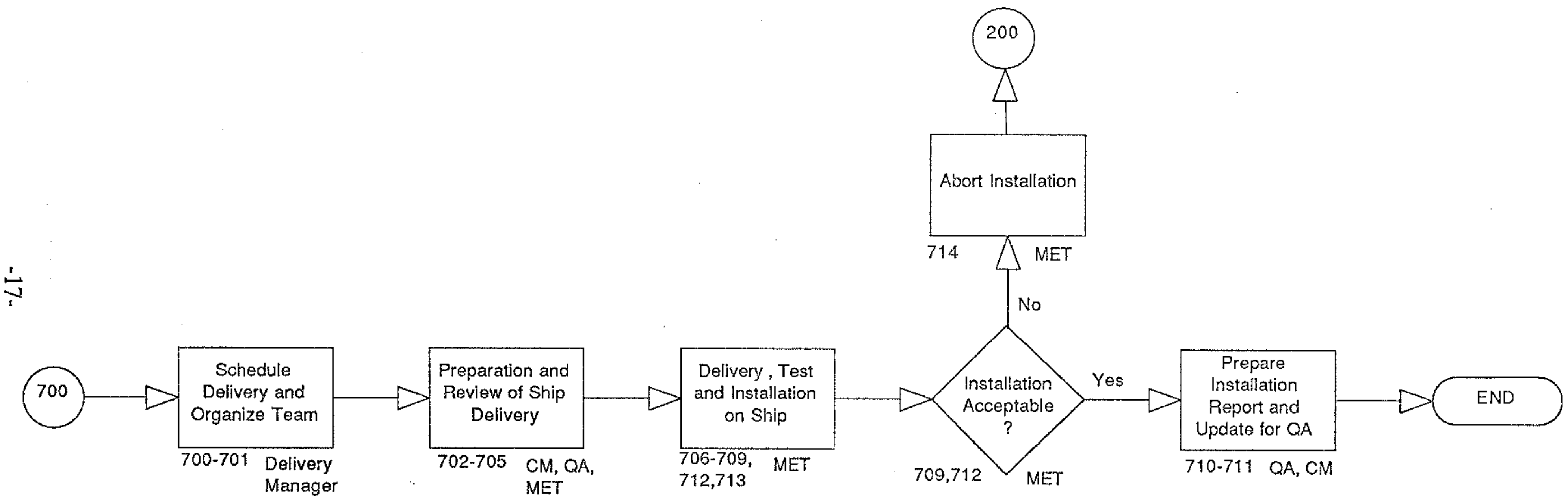

700
Schedule Delivery and Organize Team
700-701 Delivery Manager
Preparation and Review of Ship Delivery
702-705 CM, QA, MET
Delivery , Test and Installation on Ship
706-709, 712,713 MET
Installation Acceptable ?
709,712 MET
No
Abort Installation
714 MET
200
Yes
Prepare Installation Report and Update for QA
710-711 QA, CM
END

Figure 9. Delivery of Baseline Process (#700)

2.3.7 Delivery Update

Develop Update Process 800 is shown in Figure 10. An update is performed as an emergency alternative to a source code change in Process 200. This requires that a CPCR have a suitable patch solution. If one can be found, then the patch is developed and an abbreviated process is followed. The patch must be integrated, tested, and a system test performed. Finally the updated system must be delivered to ship. The need for such an emergency fix is apparently rare.

2.3.8 Analysis and Approval of Changes

The ninth and final process, CPCR Analysis and Approval (900) is presented in Figures 11a and 11b. The validity of a CPCR is established, followed by an analysis to determine the scope and nature of the problem. Following an assignment of priority (change type), the CPCR is passed through a complex review process. In this review, a change is designated as applicable to either the current or a future baseline. If applicable to the current baseline, the change could be applied as an update (patch) or upgrade (code change).

3. Alternate Modeling Perspectives

The aggregate model of the AEGIS maintenance process helps to reveal the methodological intent of the process; however, other perspectives portray varying, beneficial relationships. In this section, the aggregate model is reexamined by characterizing activities within each process as either product oriented or (maintenance) process oriented. These perspectives furnish a complementary view to that provided by the descriptions of "intent" in the aggregate model.

The product-oriented activities are those which directly affect the product (documentation or code). In contrast process-oriented activities directly affect the process.

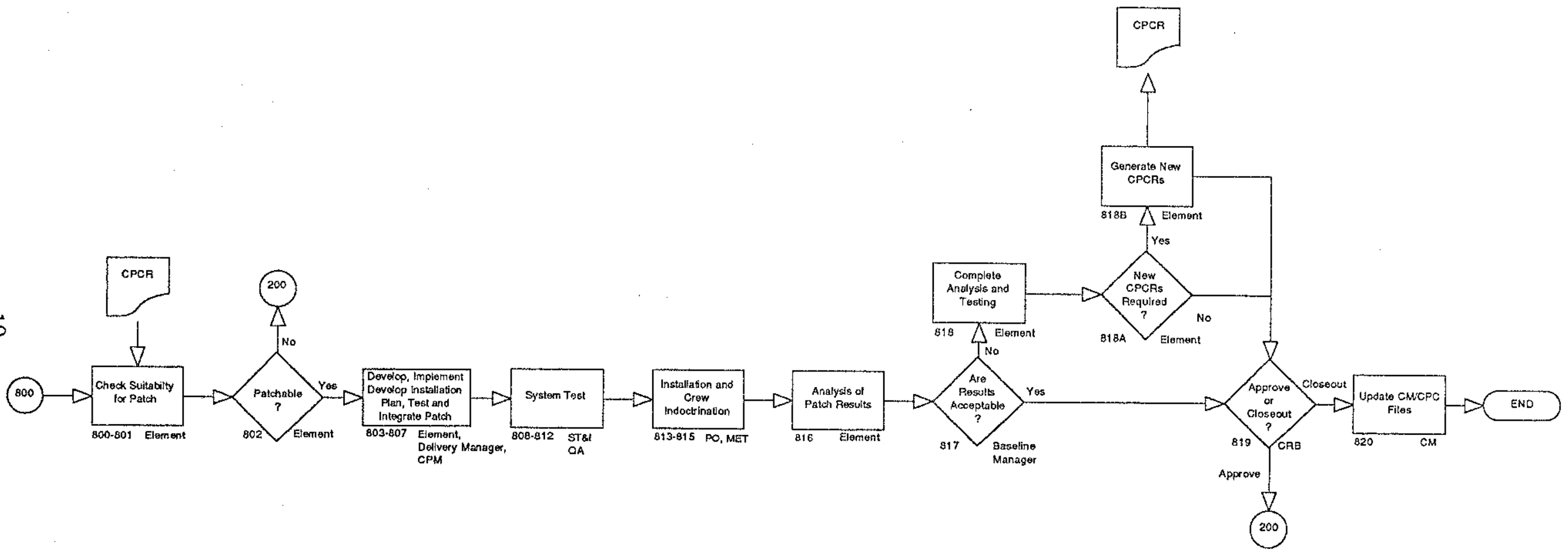


Figure 10. Develop Update Process (#800)

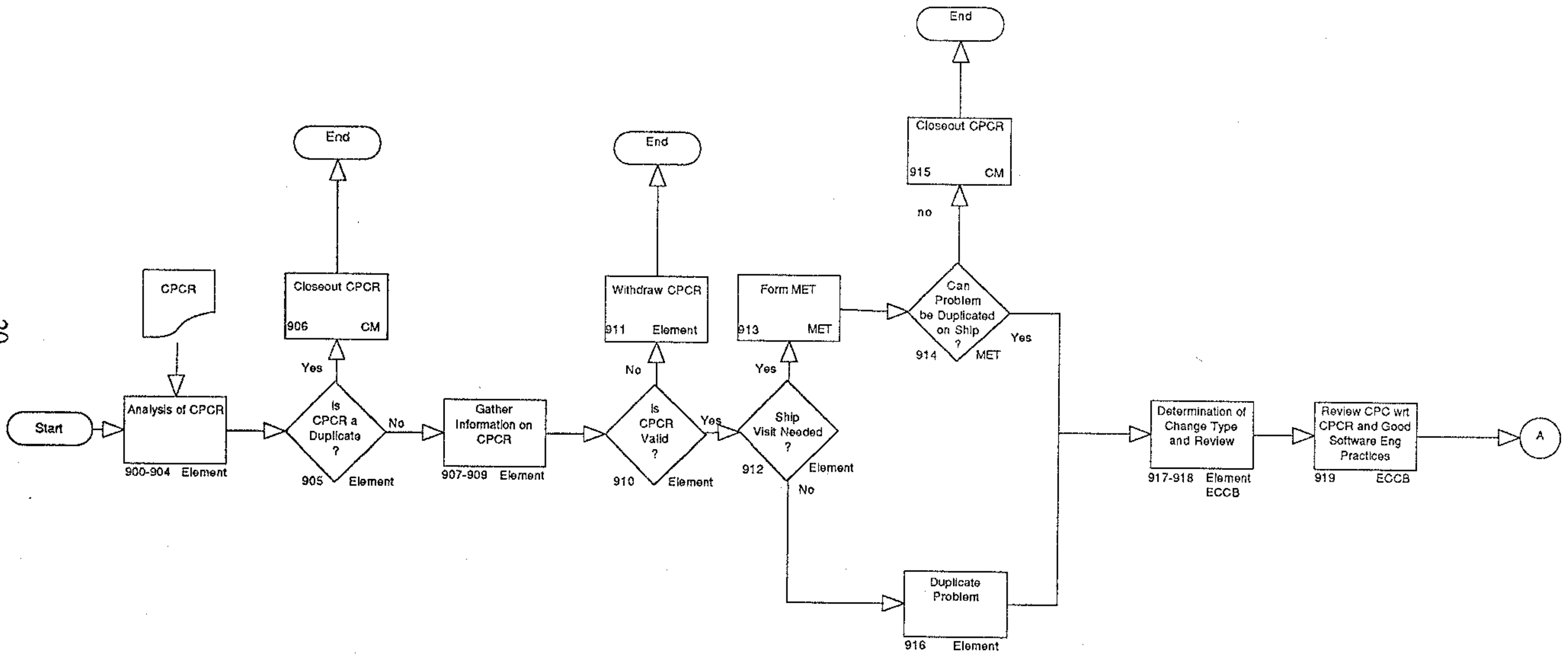


Figure 11a. A CPCR Process (#900)

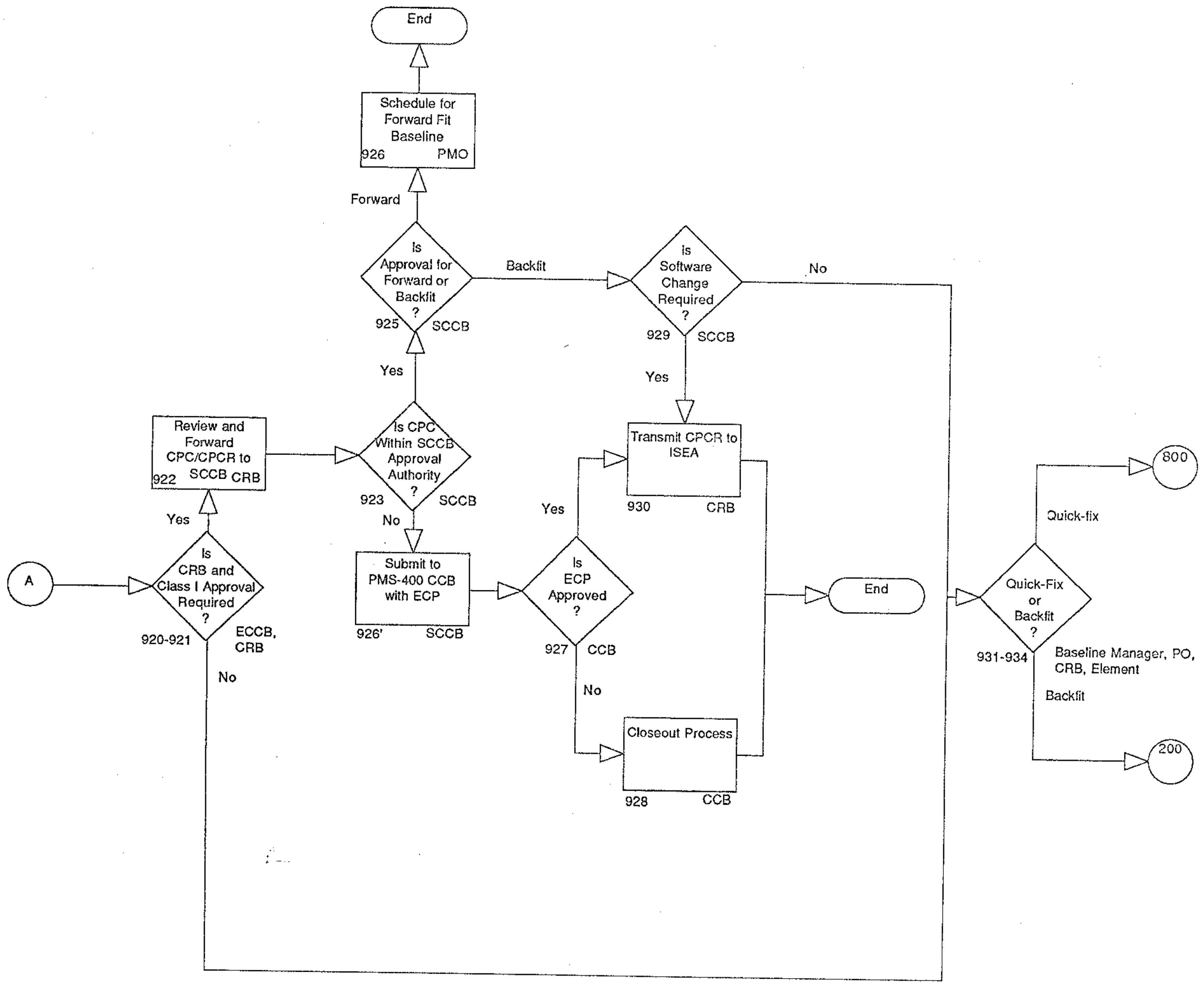

A
Is CRB and Class I Approval Required ?
920-921
ECCB, CRB
Yes
No
Review and Forward CPC/CPCR to SCCB
922
CRB
Is CPC Within SCCB Approval Authority ?
923
SCCB
Yes
No
Submit to PMS-400 CCB with ECP
926'
SCCB
Is Approval for Forward or Backfit ?
925
SCCB
Forward
Backfit
Schedule for Forward Fit Baseline
926
PMO
End
Is Software Change Required ?
929
SCCB
Yes
No
Transmit CPCR to ISEA
930
CRB
Is ECP Approved ?
927
CCB
Yes
No
Closeout Process
928
CCB
End
Quick-Fix or Backfit ?
931-934
Baseline Manager, PO, CRB, Element
Quick-fix
Backfit
800
200

Figure 11b. A CPCR Process (#900) (continuation)

While the goal of all activities is to perform software change and improvement in the most effective and efficient manner, the process oriented activities indirectly improve the product through improving the process.

The product/process division is seen elsewhere. The OPA framework, for instance, specifies mostly objectives on product quality. However, an objective for improving the maintenance process in terms of efficiency and effectiveness also exists [NANR90]. Additionally, the life-cycle model discussed in [GAMS88] is based on the division of process and product, in that both the process and product require formal definition, development and maintenance.

Finally, elements of both process and product are apparent in the activities of software development and maintenance. Activities such as specification, design and coding are all directed at producing a product. The activities of quality assurance and configuration management, however, apply directly to the process. The distinction between these perspectives seems both natural and very useful in this context.

3.1 A Product-Oriented Perspective

The majority of activities in the AEGIS process model are product-oriented. (For this reason they are not identified in the diagrams.) This conclusion indicates that the driving organizational concern is "getting the product out the door." The dominant concern with product generation can also stem from the fact that process-oriented activities are both more time consuming and costly than product-oriented ones. However, as the disciples of Total Quality Management (TQM) have noted, the ultimate, long-term determinants of quality are affected principally through process improvements.

3.2 A Process-Oriented Perspective

Process-oriented activities occur in all processes but 100, 600 and 800. In each of these, controls on the process are either unnecessary or undesirable. In the case of 800

(Develop Update), the objective is to develop and deploy the update as rapidly as possible. Although, process controls are not desirable during the majority of this process, reviews should follow deployment (so as to reflect process "health").

The existence of process-oriented activities are shown in Figures 12 through 17 by symbols with darker borders. Activities designated in this way are not entirely process-oriented; rather, some process-oriented actions are embodied within the activity. For example, the Detailed Design and Coding (200) shown in Figure 13 has four symbols indicative of process-oriented activities. The first symbol, the scheduling activity, places time constraints on all other activities; thus, affecting the process. The verification of a design review as the second symbol impacts the process. In the fourth symbol, the QA review of test plans also impacts the process. Finally, the fifth symbol includes a review of both the resulting product and the process. The remaining diagrams contain symbols darkened similarly to identify process-oriented activities.

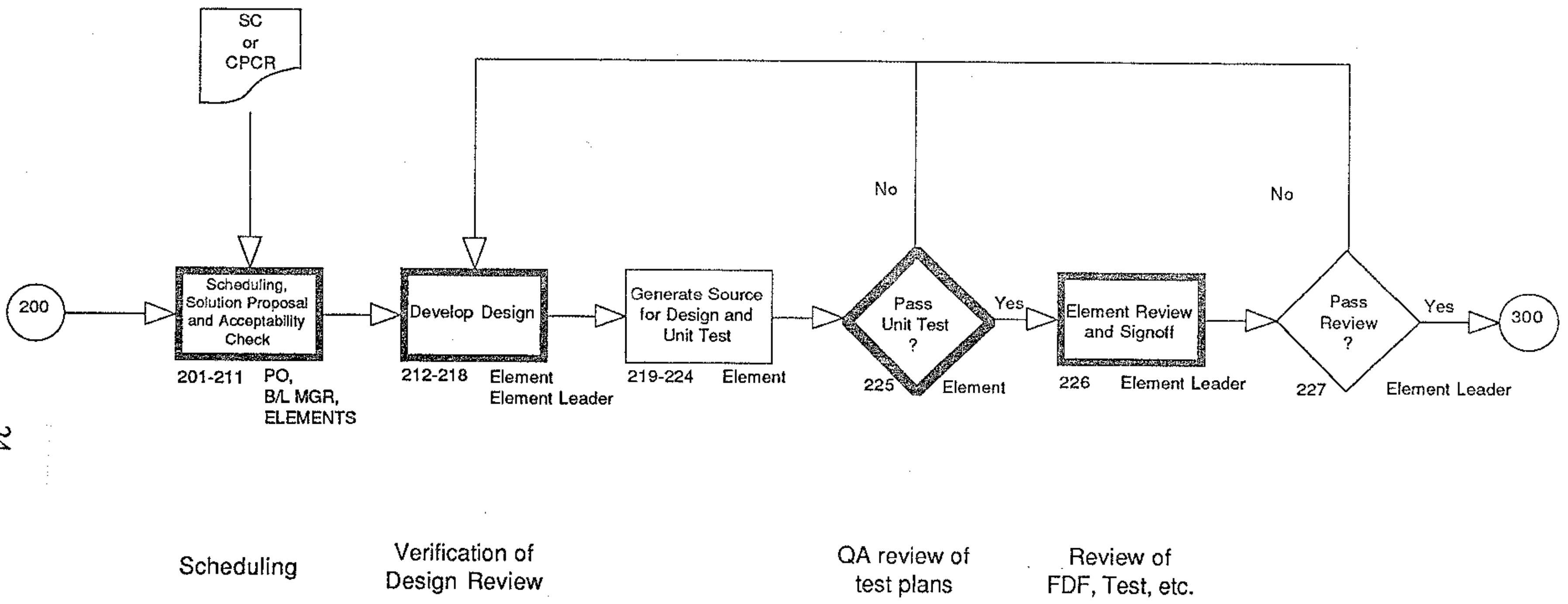
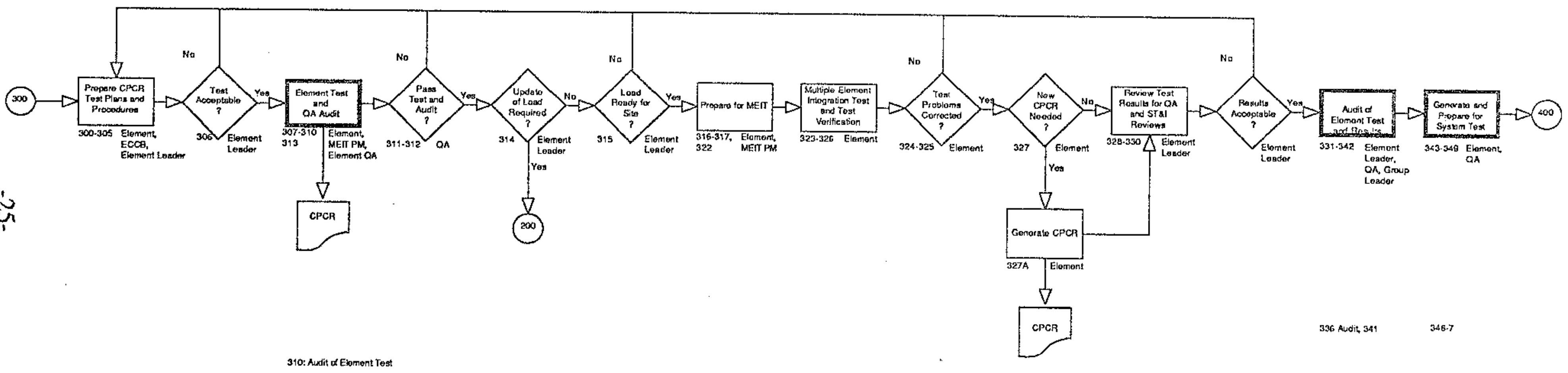

200
SC or CPCR
Scheduling, Solution Proposal and Acceptability Check
201-211 PO, B/L MGR, ELEMENTS
Develop Design
212-218 Element Element Leader
Generate Source for Design and Unit Test
219-224 Element
Pass Unit Test ?
225 Element
No
Yes
Element Review and Signoff
226 Element Leader
Pass Review ?
227 Element Leader
No
Yes
300
Scheduling
Verification of Design Review
QA review of test plans
Review of FDF, Test, etc.

Figure 12. Process-Oriented Activities of Process (#200)



300
Prepare CPCR Test Plans and Procedures
300-305 Element, ECCB, Element Leader
Test Acceptable ?
306 Element Leader
Yes
No
Element Test and QA Audit
307-310 313 Element, MEIT PM, Element QA
CPCR
310: Audit of Element Test
Pass Test and Audit ?
311-312 QA
Update of Load Required ?
314 Element Leader
200
Load Ready for Site ?
315 Element Leader
Prepare for MEIT
316-317, 322 Element, MEIT PM
Multiple Element Integration Test and Test Verification
323-326 Element
Test Problems Corrected ?
324-325 Element
New CPCR Needed ?
327 Element
Generate CPCR
327A Element
CPCR
Review Test Results for QA and ST&I Reviews
328-330 Element Leader
Results Acceptable ?
Element Leader
Audit of Element Test and Results
331-342 Element Leader, QA, Group Leader
336 Audit, 341
Generate and Prepare for System Test
343-349 Element, QA
346-7
400

Figure 13. Process-Oriented Activities of Process (#300)

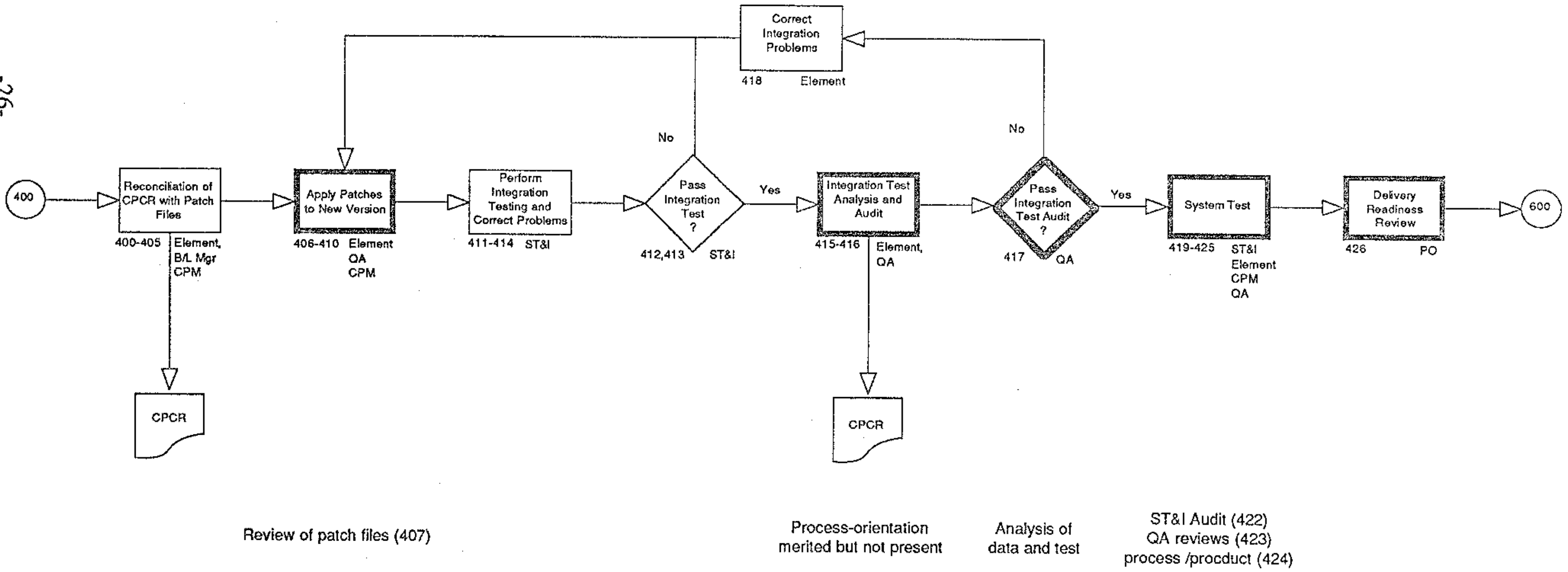

400
Reconciliation of CPCR with Patch Files
400-405
Element, B/L Mgr CPM
CPCR
Apply Patches to New Version
406-410
Element QA CPM
Perform Integration Testing and Correct Problems
411-414
ST&I
Pass Integration Test ?
412,413
ST&I
No
Yes
Correct Integration Problems
418
Element
Integration Test Analysis and Audit
415-416
Element, QA
CPCR
Pass Integration Test Audit ?
417
QA
No
Yes
System Test
419-425
ST&I Element CPM QA
Delivery Readiness Review
426
PO
600
Review of patch files (407)
Process-orientation merited but not present
Analysis of data and test
ST&I Audit (422)
QA reviews (423)
process /procduct (424)

Figure 14. Process-Oriented Activities of Process (#400)

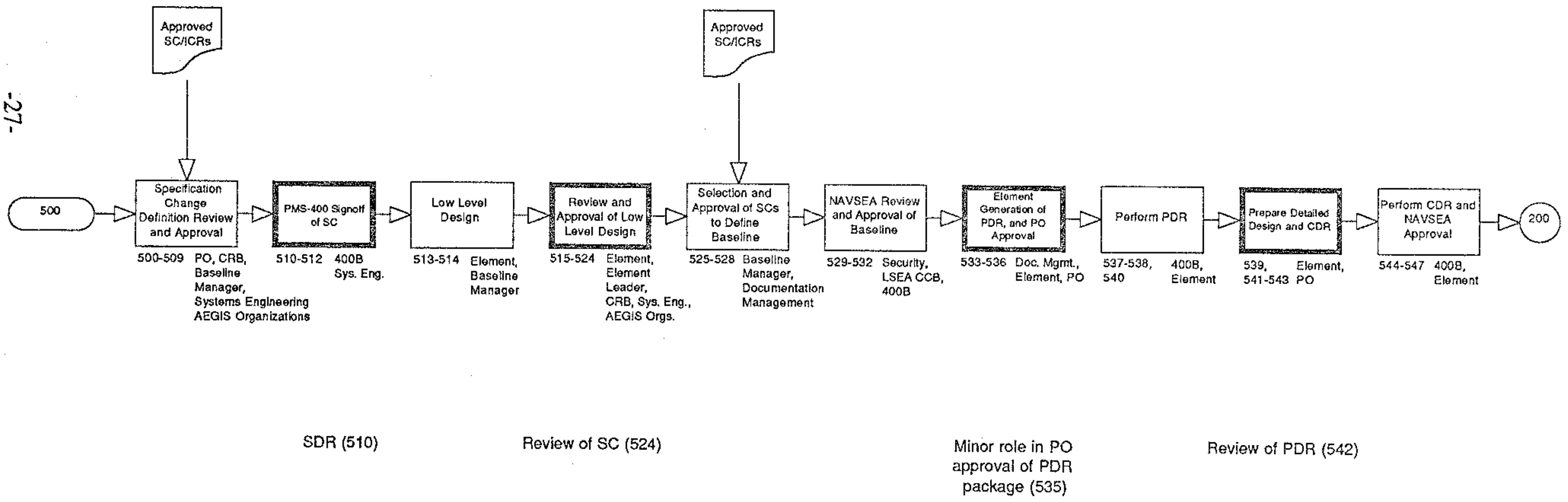

500
Approved SC/ICRs
Specification Change Definition Review and Approval
500-509 PO, CRB, Baseline Manager, Systems Engineering AEGIS Organizations
PMS-400 Signoff of SC
510-512 400B Sys. Eng.
Low Level Design
513-514 Element, Baseline Manager
Review and Approval of Low Level Design
515-524 Element, Element Leader, CRB, Sys. Eng., AEGIS Orgs.
Approved SC/ICRs
Selection and Approval of SCs to Define Baseline
525-528 Baseline Manager, Documentation Management
NAVSEA Review and Approval of Baseline
529-532 Security, LSEA CCB, 400B
Element Generation of PDR, and PO Approval
533-536 Doc. Mgmt., Element, PO
Perform PDR
537-538, 540 400B, Element
Prepare Detailed Design and CDR
539, 541-543 Element, PO
Perform CDR and NAVSEA Approval
544-547 400B, Element
200
SDR (510)
Review of SC (524)
Minor role in PO approval of PDR package (535)
Review of PDR (542)

Figure 15. Process-Oriented Activities of Process (#500)

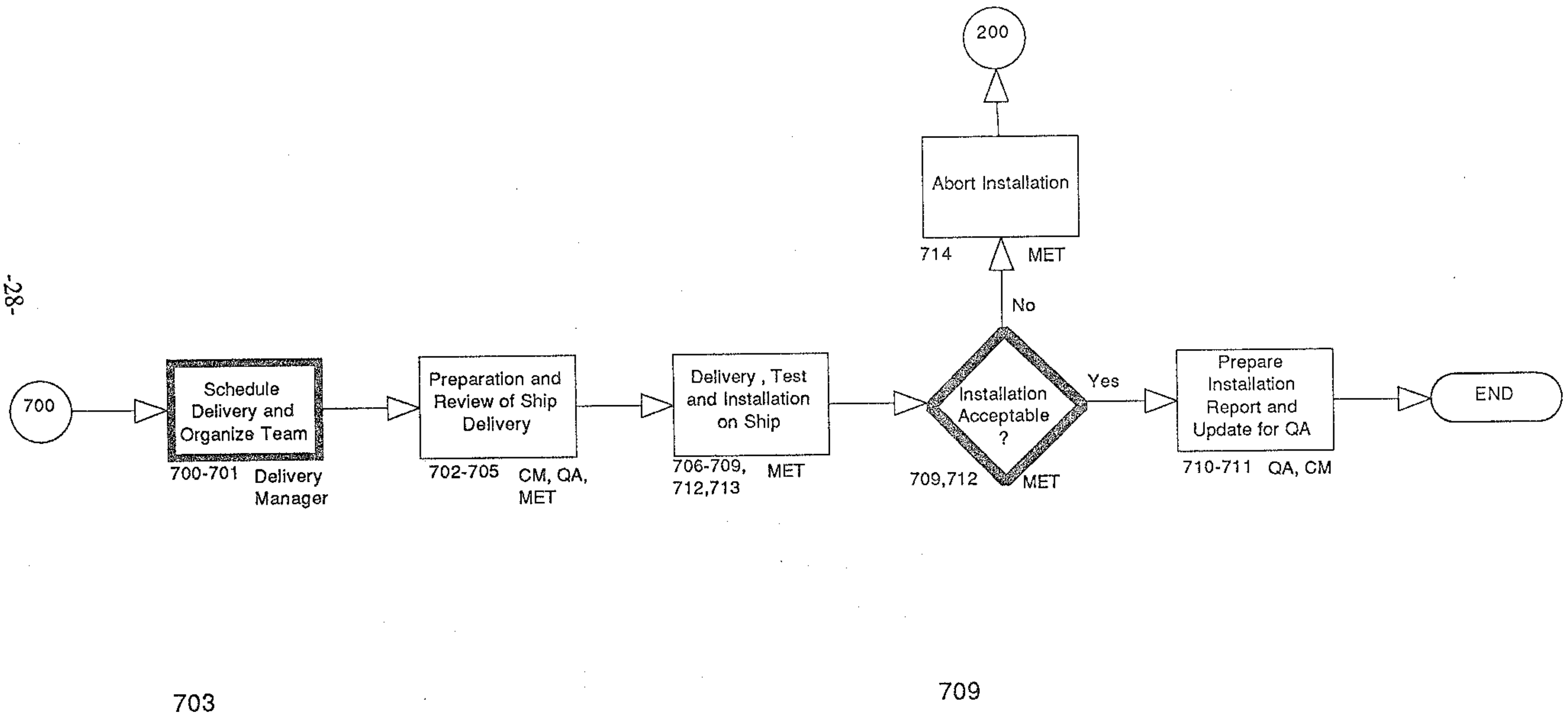

700
Schedule Delivery and Organize Team
700-701 Delivery Manager
Preparation and Review of Ship Delivery
702-705 CM, QA, MET
Delivery , Test and Installation on Ship
706-709, 712,713 MET
Installation Acceptable ?
709,712 MET
No
Abort Installation
714 MET
200
Yes
Prepare Installation Report and Update for QA
710-711 QA, CM
END
703
709

Figure 16. Process-Oriented Activities of Process (#700)

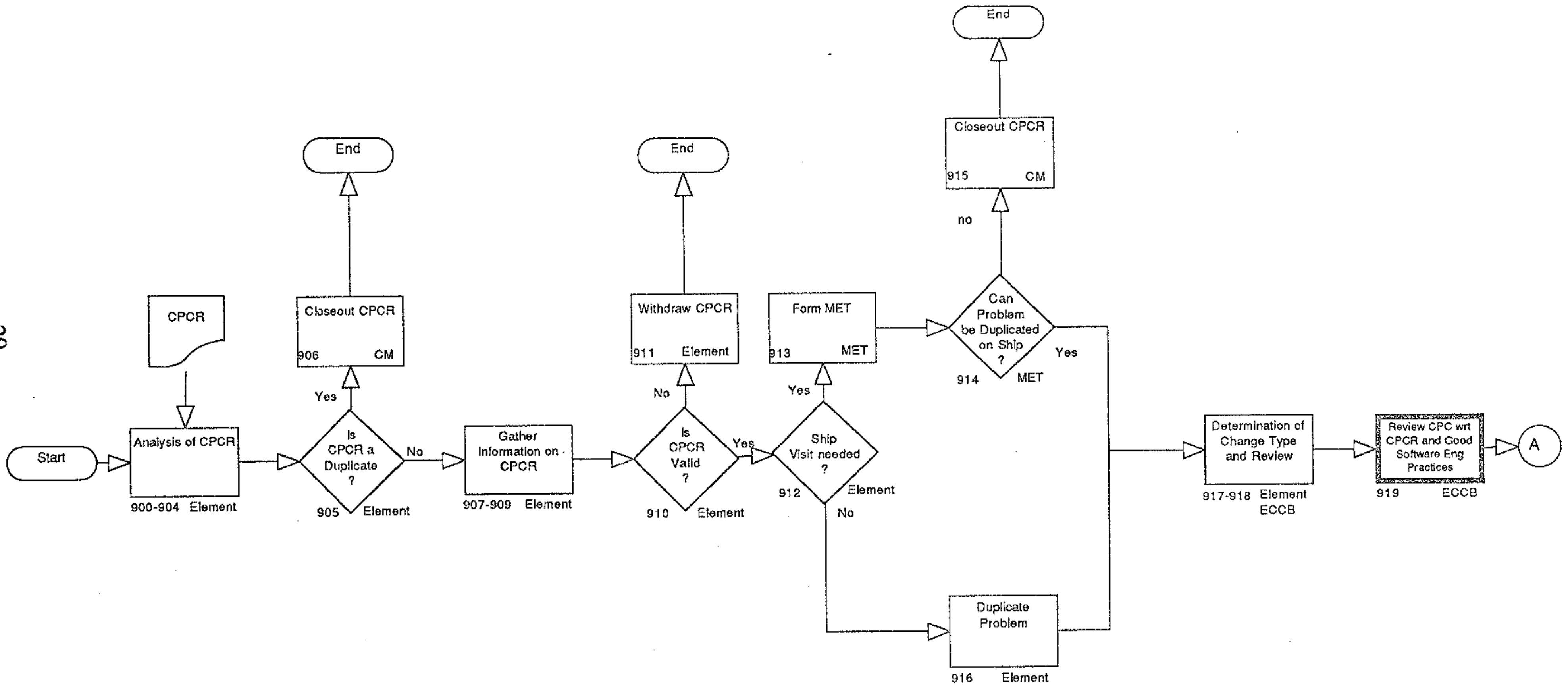
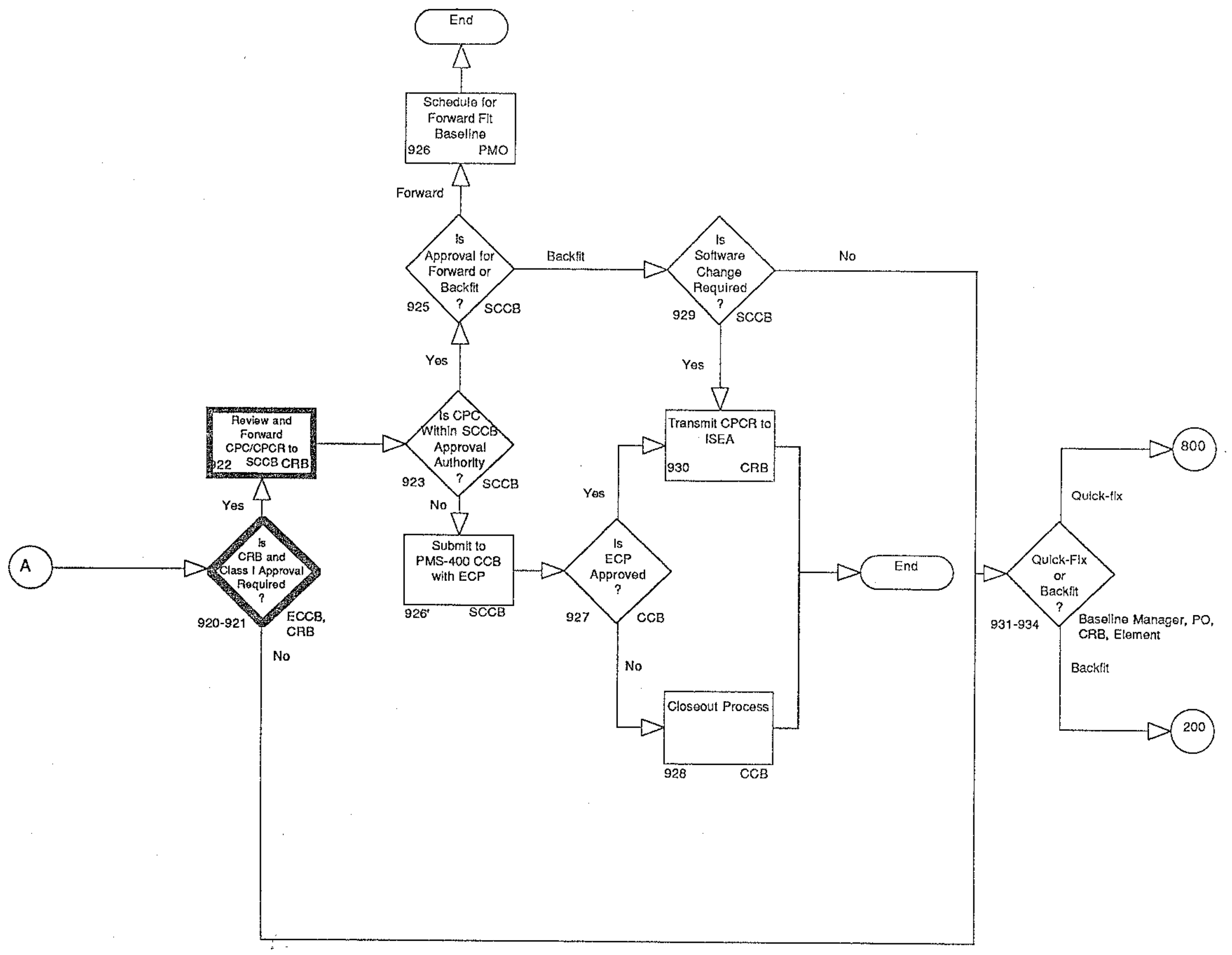


Figure 17a. Process-Oriented Activities of Process (#900)


End
Schedule for Forward Fit Baseline
926 PMO
Forward
Is Approval for Forward or Backfit ?
925 SCCB
Backfit
Is Software Change Required ?
929 SCCB
No
Yes
Review and Forward CPC/CPCR to SCCB CRB
922
Is CPC Within SCCB Approval Authority ?
923 SCCB
No
Transmit CPCR to ISEA
930 CRB
800
Quick-fix
A
Is CRB and Class I Approval Required ?
920-921 ECCB, CRB
Submit to PMS-400 CCB with ECP
926' SCCB
Is ECP Approved ?
927 CCB
End
Quick-Fix or Backfit ?
931-934
Baseline Manager, PO, CRB, Element
Backfit
No
Closeout Process
928 CCB
200

Action on the review from 919

Figure 17b. Process-Oriented Activities of Process (#900) (continuation).

4. Principle and Requirements Application

The application of the principles and requirements from Task 1 is utilized to "build" the "ideal" maintenance model. In fact, the application is used to identify those points in each process where the requirements can potentially be met. From the potential sources of improvement, the selection of the most effective or most efficient points of application can be made (Task 3).

The maintenance principles are used in the derivation of requirements; so the application of the requirements by necessity implies the application of principles which are repeated below from [NANR90, pp. 28-30].

Scope Delineation - The initiation of every task should be the identification of bounding (document) components.

Varied Abstraction - Representations that support multiple levels of abstraction and the transitions among them should be utilized in the maintenance process.

Change Propagation - Recognition of the need to propagate specification changes through multiple levels of abstraction (i.e., throughout the maintenance document set).

Quantification with Abstraction Resolution - Quantification of the product quality should be a constant goal: the potential for quantification is inversely related to the level of abstraction.

A less precise but more intuitive statement of these principles can also be given. This statement is not intended to be an alternative to replace that given above, but only to aid in the understanding of the formal statement of the principles.

Scope Delineation - Begin a maintenance task by pulling together all the documentation needed to complete it.

Varied Abstraction - Different levels of description are required to fully understand what is needed to meet maintenance objectives.

Change Propagation - Changes in code or documentation should be reflected in all the specifications (documentation) that supports the understanding and potential modification of those changes.

Quantification with Abstraction Resolution - Measurement of software quality, although difficult in early development, should be a constant goal, both to estimate maintenance effort and assess maintenance effectiveness.

4.1 Basis for Application

The requirements for a software maintenance methodology are stated in [NANR90]. They are restated below with heuristics for applying them to the process model. Note also the addition of Requirement 11, which is felt to be a necessary addition to augment and/or clarify Requirement 1.

(1) *Access to development documentation commensurate with maintenance forms.*
Principles: Varied Abstraction.
Purpose: A mechanism for accessing development documentation is the first need, but, secondly, this access should be guided by the form of maintenance. The need for accessing specific levels of documentation for particular maintenance forms is described by the Abstraction Refinement Model [NANR89].
Rule for Application: This requirement applies to any activity which utilizes development documentation.

(2) *Provide for decisions which maximize product availability (consider system availability).*
Principles: Varied Abstraction.
Purpose: This requirement relates to the global objective of the maintenance process being both effective and efficient. The methodology should seek to ensure that quality is not sacrificed but changes are accomplished efficiently.
Rule for Application: There are two ways in which this requirement might be interpreted (neither is incorrect). First, the decisions could minimize time spent in the maintenance process. Second, the decisions could maximize product quality, and in particular reliability. Both should be combined since neither alone is adequate. However, the application of this requirement is necessary where decisions of either type are made.

(3) *Each modification activity should include the following sub-activities:*

(a) Identify source and target (order is source dependent).

(b) Define and effect transformation process.

(c) Record source, process, and target.

(d) Test:

(1) Identify original test specifications

(2) Modify original test specifications for target.

(3) Revise test procedures and apply them.

Principles: Varied Abstraction, Scope Delineation, Change Propagation.

Purpose: The major steps in making a modification are prescribed. Actually making the modification requires the identification of the source of the maintenance actions, the target of the actions (a solution) and the means for realizing that target. Complementing these activities is the recording of the source, target and the process (including design decisions) to ensure compatibility of the documentation with the programs. Finally, testing is required to ensure that the correct target is achieved.

Rule for Application: This requirement governs the modifications made during maintenance, and so impacts these activities. It is not necessary that each of the parts of the requirement occur in sequence with no separating activities.

(4) *Promote the identification of alternatives, the evaluation of alternatives (risk assessment), and support documentation of both.*

Principles: Varied Abstraction.

Purpose: The maintenance (and development) activities involve the consideration of a number of alternatives. The recording of these alternatives and their evaluation provide a justification of the selected change strategy that may be useful for later maintenance.

Rule for Application: Many alternatives arise in maintenance. The ones governed by this requirement should be those which impact the system at a lower level of decision making. In general, these would be alternatives with more than two choices and would not be of the nature "does the system pass this test?"

(5) *Recognize and resolve potential interference among concurrent maintenance activities.*
Principles: Scope Delineation.
Purpose: Maintenance activities might require changes which interfere or interact in some way. Recognizing and handling this interference helps ensure that combinations of modifications have a positive result. Otherwise, these modifications might combine to form an undesired change or one that is obviated by a subsequent change.
Rule for Application: This requirement applies at the initial consideration of a modification.

(6) *Require and facilitate auditing of the maintenance process (metrics, and methodology).*
Principles: Quantification with Abstraction Refinement.
Purpose: An audit of the maintenance process evaluates the success of the process in terms of the product quality. The methodology should support auditing to guarantee the correctness of the process and the quality of the product.
Rule for Application: This requirement impacts all activities which involve preparation for and evaluation of the maintenance process.

(7) *Support (enforce) uniformity in maintenance process/activity (procedures, documentation).*
Principles: Change Propagation, Varied Abstraction.
Purpose: Providing for uniformity in the maintenance process means that the procedures used in the maintenance process are the same. The impact of uniformity

in the process is uniformity in documentation. Uniformity in documentation increases the maintainability of the system by facilitating understanding of the system.

Rule for Application: This requirement should apply to all of the activities of the process.

(8) *Enable prioritization and coordination of maintenance forms and activities.*

Principles: Change Propagation.

Purpose: Various maintenance forms attach not only a practical order, but also a theoretical order to tasks. The practical order captures the necessity for certain modifications being made first. The theoretical order implies that certain tasks should be performed first for efficiency and effectiveness. The methodology needs to recognize the ordering in the decision-making process.

Rule for Application: This requirement influences planning and scheduling activities, but also the initial consideration of a problem (i.e. CPCR).

(9) *Enforce recording of source, process, target, test documentation, decision alternatives, evaluation, and final decision.*

Principles: Scope Delineation, Change Propagation.

Purpose: To provide for the documentation necessary for future maintenance.

Rule for Application: This requirement enforces the recording of information from a variety of sources including modifications, design decisions, and testing.

(10) *Enable, promote and enforce the quantification of the process and product quality.*

Principles: Quantification with Abstraction Resolution.

Purpose: Metrics can play a major role in the auditing process if provided proper support.

Rule for Application: This requirement impacts all activities.

(11) *Facilitate access to documentation that reflects the impact of changes made during maintenance.*

Principles: Scope Delineation, Varied Abstraction, Change Propagation

Purpose: Documentation access is a critical aspect of the planning process for maintenance. While Requirement 1 establishes the necessity of a means for accessing development documentation according to the level of abstraction, this requirement is intended to deal with the fact that development documentation is gradually replaced by documentation which reflects the activities of maintenance on the system.

Rule for Application: This requirement impacts all activities which utilize documentation of any form. In general, these are the planning and analysis activities for modification and testing.

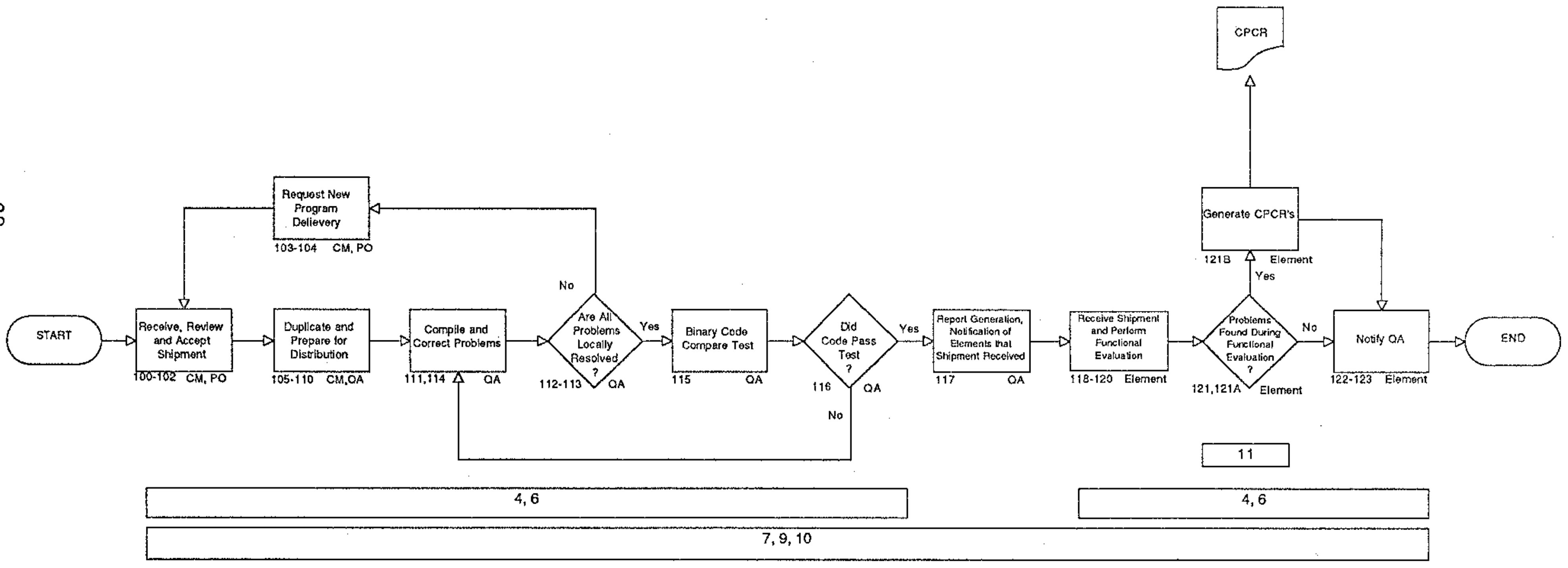
The relationship between principles and requirements is shown in Table 2. Entries in the table indicate whether the principles and requirements impact activities which are process-oriented, product-oriented or both.

Table 2. Relationship of Principles and Requirements Shown by Process or Product Impact

Requirement	Principle: Scope Delineation	Varied Abstraction	Quantification w/Abstraction Resolution	Change Propagation
(1) Documentation Access vs. Forms		Product		
(2) Product Availability		Process		
(3) Activity Definition	Process/ Product	Process/ Product		Product
(4) Treatment of Alternatives		Process		
(5) Task Interference	Process			
(6) Process Auditing			Process/ Product	
(7) Process Uniformity		Process		Process/ Product
(8) Maintenance Form Order				Process
(9) Record Keeping	Process/ Product		Product	Process/ Product
(10) Quantification of Quality			Process/ Product	
(11) Documentation Access	Process	Process		

4.2 Requirements Characterization of Maintenance Activities

The application of the requirements to the AEGIS process model is shown in Figures 18 through 26. Each numbered bar underscores the activities to which the requirement applies. This designation means only that the requirement should be applied to that activity, not that the requirement is already met; although, in some cases the requirement is met by the corresponding activity. Rather than discuss every diagram, each requirement is described for at least one diagram.



START
Receive, Review and Accept Shipment
100-102 CM, PO
Request New Program Delievery
103-104 CM, PO
Duplicate and Prepare for Distribution
105-110 CM,QA
Compile and Correct Problems
111,114 QA
No
Are All Problems Locally Resolved ?
112-113 QA
Yes
Binary Code Compare Test
115 QA
Did Code Pass Test ?
116 QA
No
Yes
Report Generation, Notification of Elements that Shipment Received
117 QA
Receive Shipment and Perform Functional Evaluation
118-120 Element
Problems Found During Functional Evaluation ?
121,121A Element
Yes
Generate CPCR's
121B Element
CPCR
No
Notify QA
122-123 Element
END
11
4, 6
4, 6
7, 9, 10

Figure 18. Requirements Application to Process (#100)

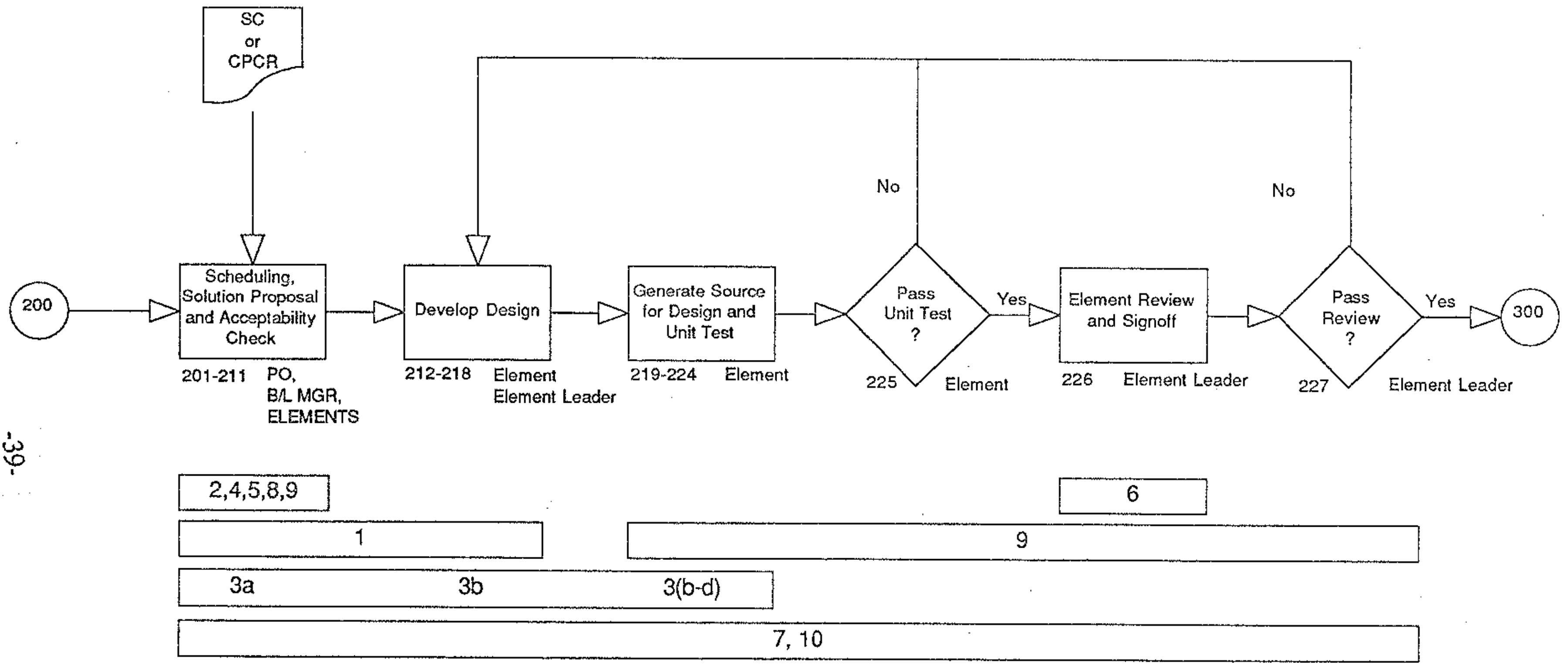


Figure 19. Requirements Application to Process (#200)

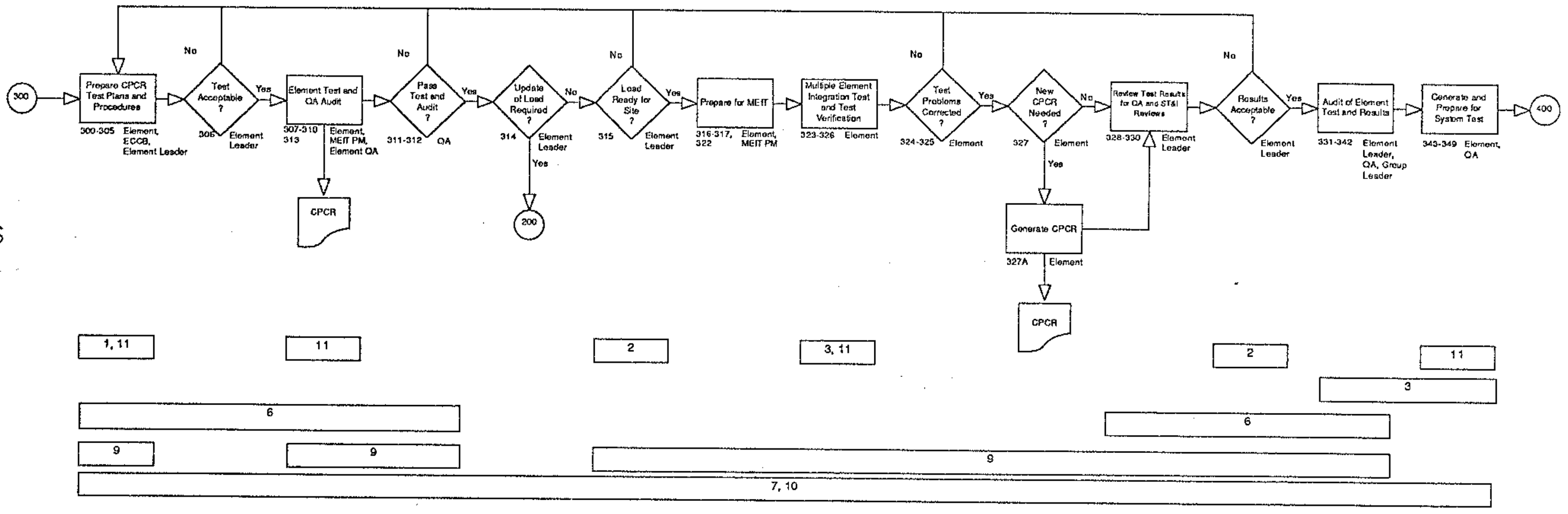

300
Prepare CPCR Test Plans and Procedures
300-305 Element, ECCB, Element Leader
Test Acceptable ?
306 Element Leader
No
Yes
Element Test and QA Audit
307-310 313 Element, MEIT PM, Element QA
CPCR
Pass Test and Audit ?
311-312 QA
No
Yes
Update of Load Required ?
314 Element Leader
No
Yes
200
Load Ready for Site ?
315 Element Leader
No
Yes
Prepare for MEIT
316-317, 322 Element, MEIT PM
Multiple Element Integration Test and Test Verification
323-326 Element
Test Problems Corrected ?
324-325 Element
No
Yes
New CPCR Needed ?
327 Element
No
Yes
Generate CPCR
327A Element
CPCR
Review Test Results for QA and ST&I Reviews
328-330 Element Leader
Results Acceptable ?
Element Leader
No
Yes
Audit of Element Test and Results
331-342 Element Leader, QA, Group Leader
Generate and Prepare for System Test
343-349 Element, QA
400
1, 11
11
2
3, 11
2
11
3
6
6
9
9
9
7, 10

Figure 20. Requirements Application to Process (#300)

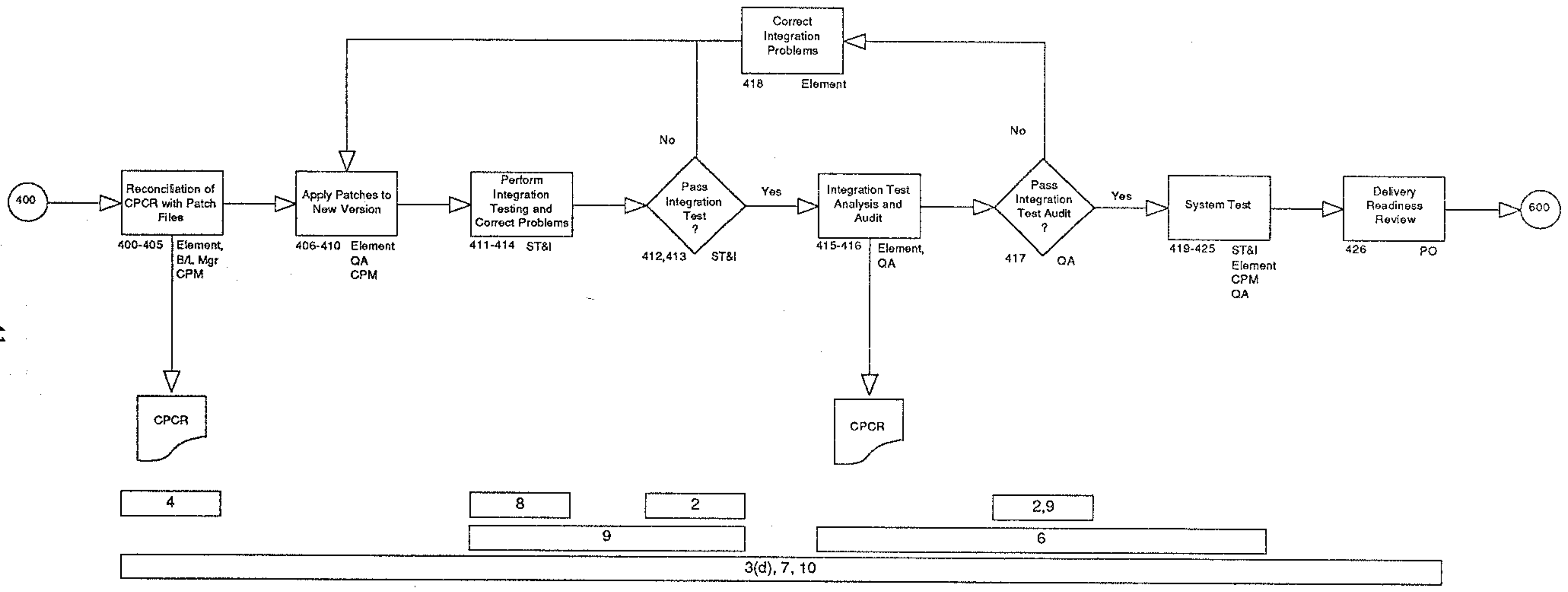

400
Reconciliation of CPCR with Patch Files
400-405
Element, B/L Mgr CPM
CPCR
Apply Patches to New Version
406-410
Element QA CPM
Perform Integration Testing and Correct Problems
411-414
ST&I
Pass Integration Test ?
412,413
ST&I
No
Yes
Correct Integration Problems
418
Element
Integration Test Analysis and Audit
415-416
Element, QA
CPCR
Pass Integration Test Audit ?
417
QA
No
Yes
System Test
419-425
ST&I Element CPM QA
Delivery Readiness Review
426
PO
600
4
8
2
2,9
9
6
3(d), 7, 10

Figure 21. Requirements Application to Process (#400)

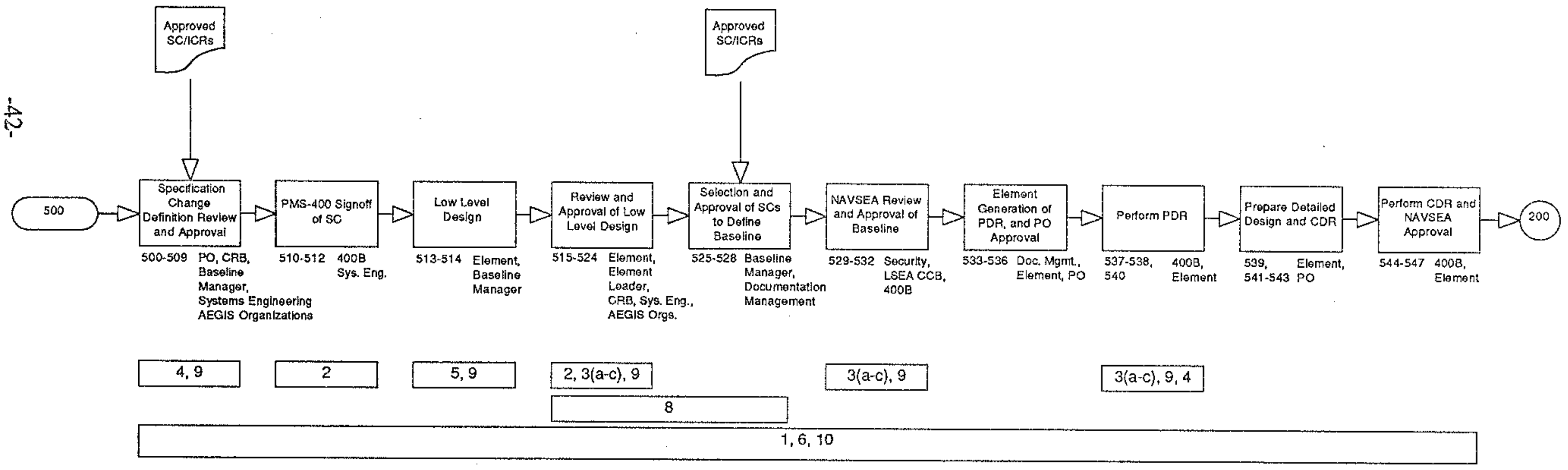

500
Approved SC/ICRs
Specification Change Definition Review and Approval
500-509 PO, CRB, Baseline Manager, Systems Engineering AEGIS Organizations
PMS-400 Signoff of SC
510-512 400B Sys. Eng.
Low Level Design
513-514 Element, Baseline Manager
Review and Approval of Low Level Design
515-524 Element, Element Leader, CRB, Sys. Eng., AEGIS Orgs.
Approved SC/ICRs
Selection and Approval of SCs to Define Baseline
525-528 Baseline Manager, Documentation Management
NAVSEA Review and Approval of Baseline
529-532 Security, LSEA CCB, 400B
Element Generation of PDR, and PO Approval
533-536 Doc. Mgmt., Element, PO
Perform PDR
537-538, 540 400B, Element
Prepare Detailed Design and CDR
539, 541-543 Element, PO
Perform CDR and NAVSEA Approval
544-547 400B, Element
200
4, 9
2
5, 9
2, 3(a-c), 9
3(a-c), 9
3(a-c), 9, 4
8
1, 6, 10

Figure 22. Requirements Application to Process (#500)

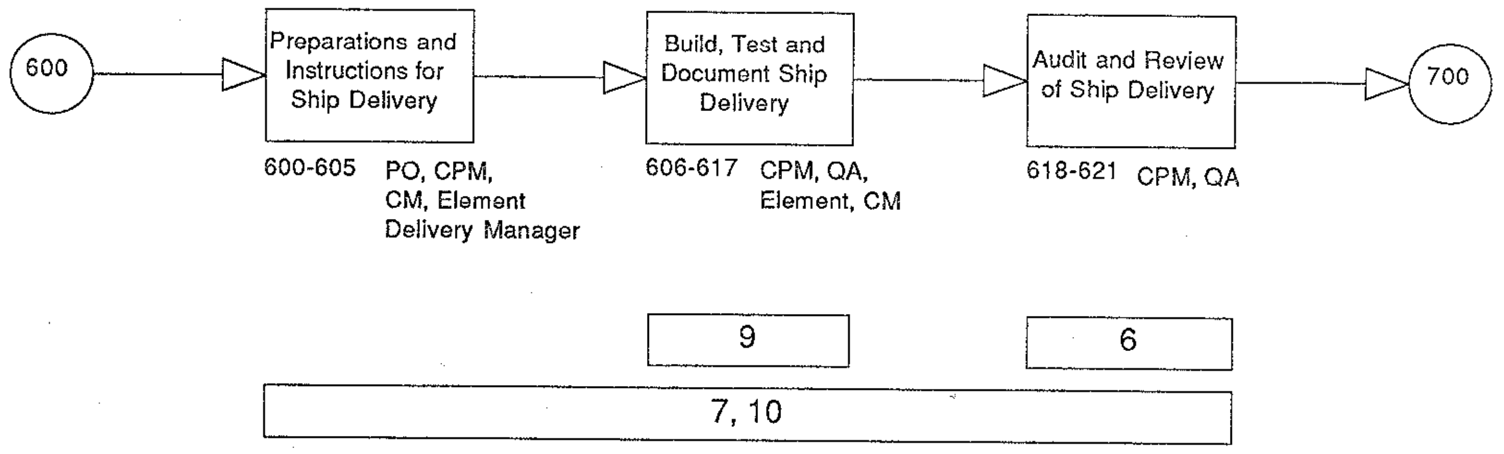

600
Preparations and Instructions for Ship Delivery
600-605 PO, CPM, CM, Element Delivery Manager
Build, Test and Document Ship Delivery
606-617 CPM, QA, Element, CM
Audit and Review of Ship Delivery
618-621 CPM, QA
700
9
6
7, 10

Figure 23. Requirements Application to Process (#600)

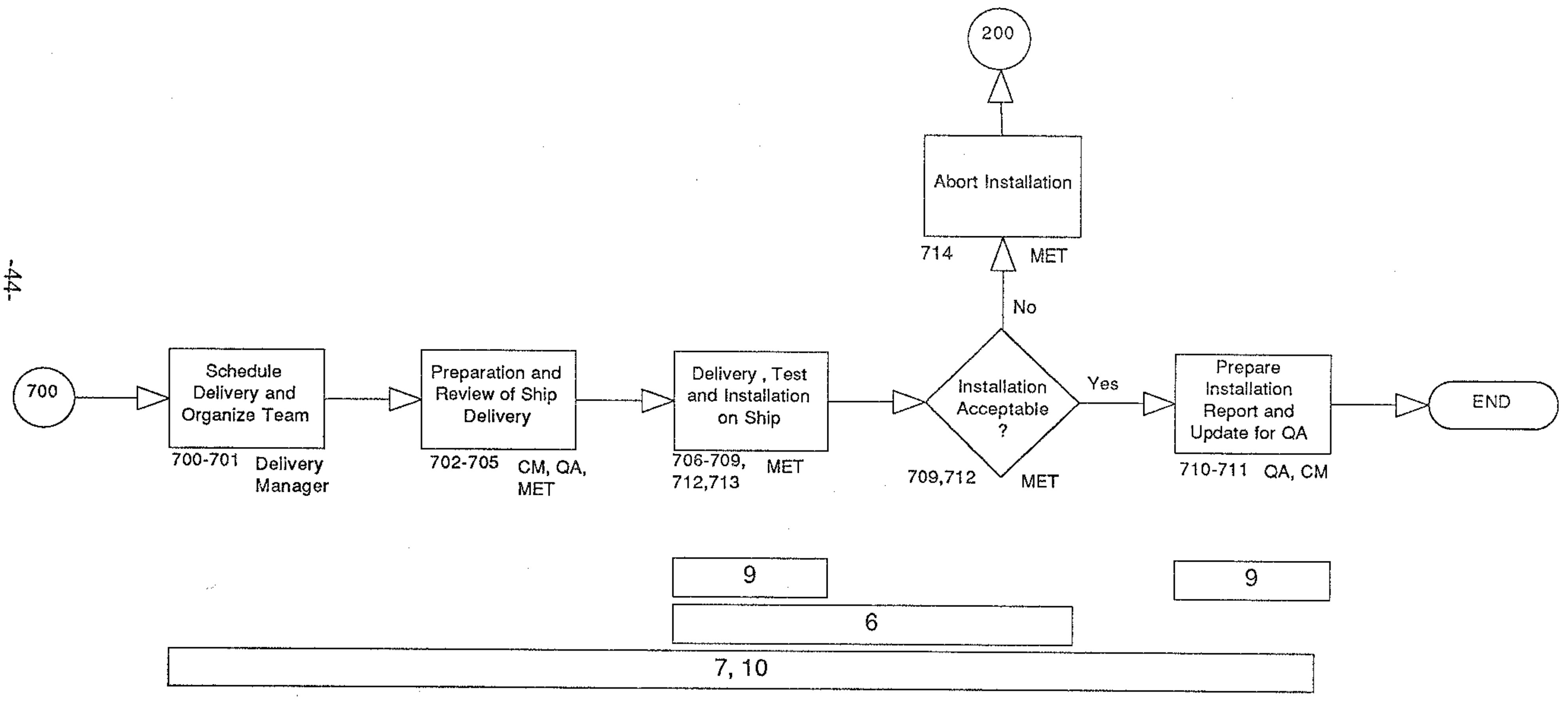

700
Schedule Delivery and Organize Team
700-701 Delivery Manager
Preparation and Review of Ship Delivery
702-705 CM, QA, MET
Delivery , Test and Installation on Ship
706-709, 712,713 MET
Installation Acceptable ?
709,712 MET
No
Abort Installation
714 MET
200
Yes
Prepare Installation Report and Update for QA
710-711 QA, CM
END
9
6
7, 10
9

Figure 24. Requirements Application to Process (#700)

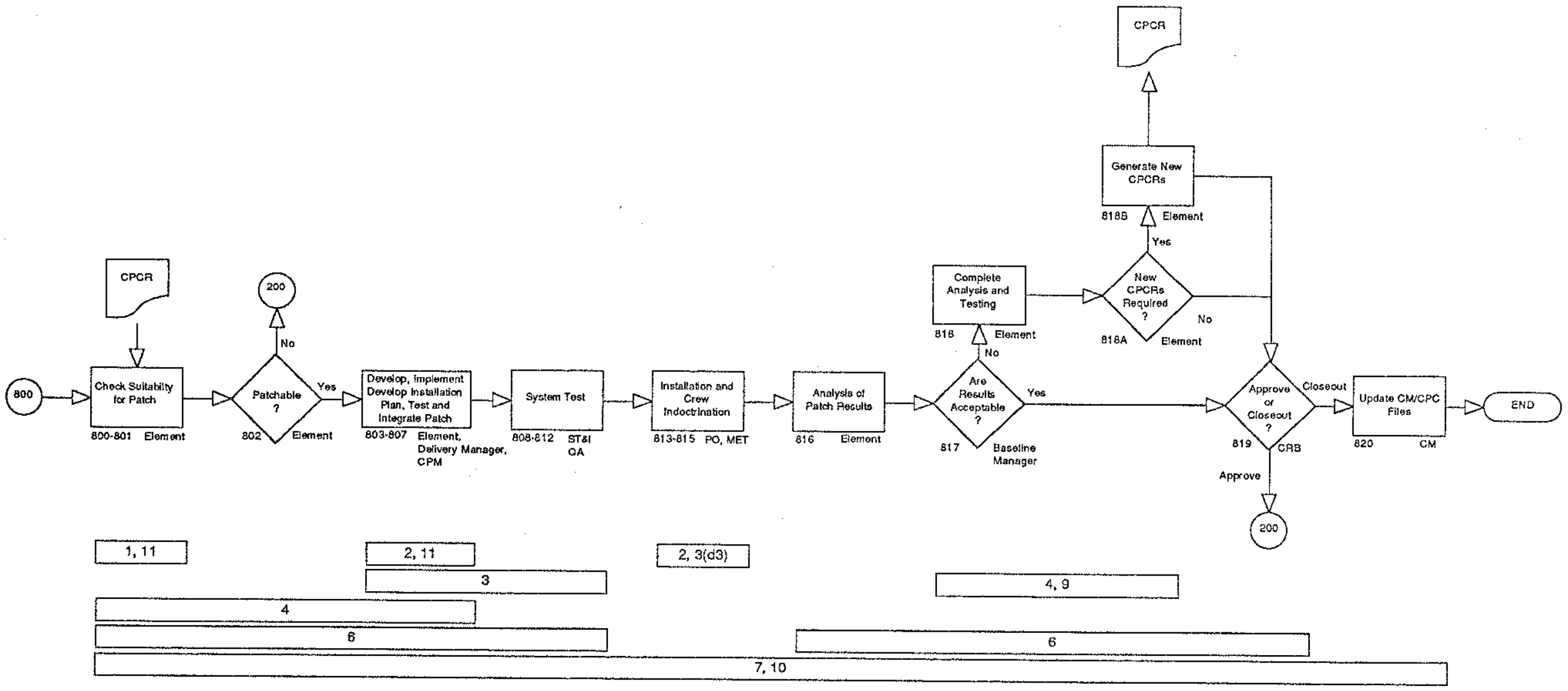


Figure 25. Requirements Application to Process (#800)

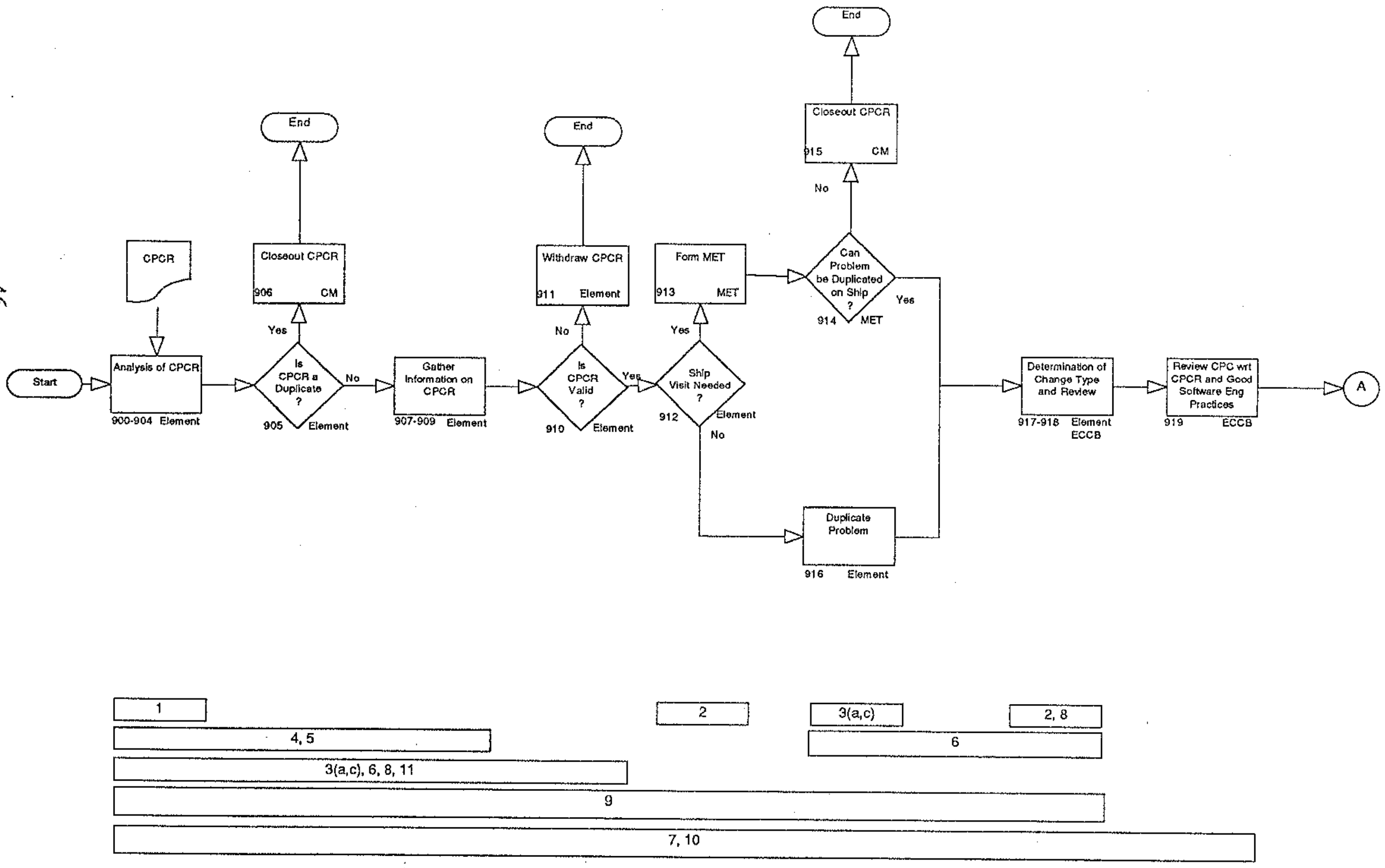


Figure 26a. Requirements Application to Process (#900)

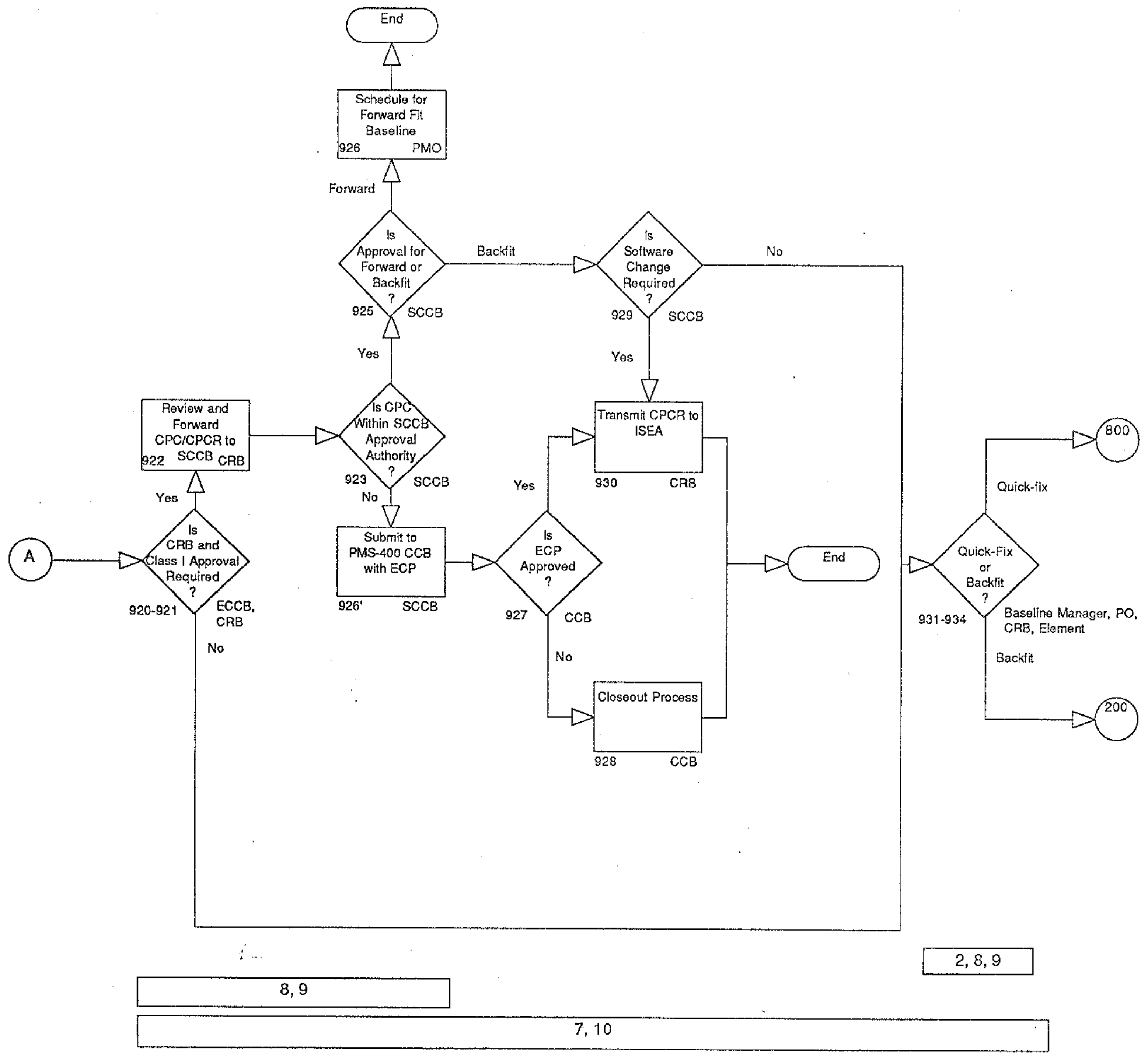

A
Is CRB and Class I Approval Required ?
920-921
ECCB, CRB
Yes
No
Review and Forward CPC/CPCR to SCCB
922
CRB
Is CPC Within SCCB Approval Authority ?
923
SCCB
Yes
No
Submit to PMS-400 CCB with ECP
926'
SCCB
Is Approval for Forward or Backfit ?
925
SCCB
Forward
Backfit
Schedule for Forward Fit Baseline
926
PMO
End
Is Software Change Required ?
929
SCCB
Yes
No
Transmit CPCR to ISEA
930
CRB
Is ECP Approved ?
927
CCB
Yes
No
Closeout Process
928
CCB
End
Quick-Fix or Backfit ?
931-934
Baseline Manager, PO, CRB, Element
Quick-fix
Backfit
800
200
8, 9
7, 10
2, 8, 9

Figure 26b. Requirements Application to Process (#900) (continuation)

In Process 100, only Requirements 4, 6, 7, 9, 10 and 11 are applicable. Requirements 4 and 6 are applied to all but one activity. For the former (4), application is made to all activities since each could involve the consideration of alternatives regarding the acceptance and handling of the shipment. The latter, Requirement 6, is applicable because all future audits are to be made against the new baseline, and establishment of the basis for these audits is necessary upon receipt. As stated earlier, Requirements 7 and 10 are applied to all activities. Requirement 9 is applied to all activities because records of the receipt and evaluation have some impact on future maintenance. Requirement 11 is applied because the generation of a CPCR may be in recognition of a deviation from the requirements.

In Process 200 nearly all requirements are applied. Requirement 1 is applicable to the first two activities because access to development documentation may be necessary in the planning and design activities. The scheduling aspect of the first activity is governed by Requirement 2, since scheduling affects system availability. Requirement 3 applies to the first three activities. In particular, these activities are structured similarly to the modification scheme given by Requirement 3. Requirement 5 impacts the first activity since potential interference among maintenance tasks should be recognized at this early stage of planning. In addition, the planning stage of modifications requires recognition of the priorities of the maintenance forms and their impact on the order of maintenance tasks.

The requirements have been applied to the remainder of the activities comprising the process representations in a similar fashion. For most, the justification for application is given in the previous section.

4.3 Implications for an "Ideal" Model

The application of the requirements to the aggregated AEGIS Maintenance Process gives a hint of the improvements which could be made to progress toward an "ideal" model. For each process, the designation of activities to which the requirement applies suggests the potential for improving the activity.

For example, in Process 200, Requirement 1 is applied to the first two activities. In each it should be recognized that appropriate levels of documentation are the basis of the task and that the means for accessing this documentation should be provided. In addition, the application of Requirement 5 suggests that the activity should account for other maintenance tasks which are being processed concurrently. For each of these requirements the corresponding activities could be improved in a manner directed by the requirement. Some coordination between the respective improvements is needed to result in a coherent process and an "ideal" or improved model.

5. Summary

The relationship between maintenance methodology and process is an important tool in the development of a methodology. This report describes two intermediate models derived from the detailed (current) model. The software maintenance principles and requirements are applied to an aggregated model of "intent" to derive an improved model of the AEGIS Maintenance Process. This "ideal" model is part of the basis for Task 3, in which an AEGIS Maintenance Methodology is to be defined.

Recommendations and directions suggested by Task 2 are presented below to highlight the findings of Task 2. The directions present the anticipated role the "ideal" model is to play in the development of the AEGIS Maintenance Methodology.

5.1 Recommendations

The following recommendations are based on observations and concerns regarding the execution of Task 2.

5.1.1 Recognize Limitations of Process Representation

It is suggested in section 2.2 that the flow chart representation of the AEGIS maintenance process may be limiting. The flow chart is suitable for description of the procedural aspects of the process, but does not show the relationships of the process to the product. Because of the sequential nature of the flow chart, the inherent parallelism in the process is not apparent, and some activities may be mis-represented. For instance, the Detailed Design and Coding activity (200) may be activated for numerous CPCR's before Element Test is initiated, possibly with several instantiations "executing" concurrently.

5.1.1.1 Implications for Methodology Definition

The implications for the definition of the methodology are substantial. The methodology may be based on a model which incorrectly represents the current process. The improved model of this process suffers by unnecessarily imposing a sequential structure on most activities. Parallel activities such as design and system test planning are forced into a sequential order.

Although, the limitations of the model can be recognized and perhaps dealt with appropriately, the direction provided by the model could be lost. Clearly, the requirements application can aid the identification of improvements to each activity, but the task ordering provided by the model is also an important aspect of the methodology. The flow chart representation limits the way in which the activities may relate.

Without parallelism in the process, some maintenance requirements cannot be met (in particular Requirement 2). Certainly process-oriented activities should be carried out in parallel with the associated product-oriented activities. The lack of explicit parallelism either forces sequentiality on the process, or, more likely, contributes to a haphazard approach to representing parallel and sequential activities.

5.1.1.2 Alternate Formalisms

The data flow diagram has been suggested as an alternative to the flow chart. The functional aspects of the major activities of the AEGIS Maintenance Process are indicated in Figure 2 in Section 2.2. Despite the incompleteness of the diagram, the parallel nature of the process is better represented. A structured analysis approach might prove useful in the development of such a model, although the flow chart model would also prove valuable in this effort.

5.1.2 Improve Balance of Process/Product Orientation

The identification of process-related subactivities in Figures 12-17 notes a general imbalance in the concern for quality of the product – at the expense of the *maintenance process* improvement. Continued monitoring of process quality, especially as a responsibility of the SQA organization, is mandatory for the long-term advances in product quality.

5.1.3 Enforce and Support Recording of Alternatives and Decisions

Cited as the missing component for assuring effective and efficient maintenance in all too many cases [BUNG90], the documentation of alternative choices and the rationale underlying design decisions should be enforced in the deliverables from the development process. Modifications during the maintenance phase should be subject to the same level of enforcement and supported by tools to accommodate the preservation of this information.

5.1.4 Develop and Implement Quality Metrics

Quantification of both product and process quality is essential to the management of the AEGIS software maintenance process. Much remains to be learned about the measurement of software quality, and absolute measures may never be possible. Nevertheless, a metrification program that enables the determination of relative differences provides a basis for gauging improvements, identifying process problem spots, and setting meaningful quality goals. In the absence of metrification, effective quality improvements are difficult to substantiate and demonstrate.

5.1.5 Select and Integrate Maintenance Tools

Consistent with the OPA framework is the derivation of tool requirements from the principles that govern the maintenance process [NANR88]. In the AEGIS application domain, the enuniciation of methodology requirements (see Section 4.1) furnishes a more definitive specification of tool requirements. A key issue is the use of the maintenance methodology principles to effect an *integrated* toolset (an *environment*).

5.1.6 Rebuild Aggregate Model

The final recommendation regards reuniting the efforts reported here and the definition of the maintenance process which is on-going at NAVSWC. The aggregate and "ideal" models presented are based on older versions of the process flow diagrams. The result is that some significant differences may exist between the "actual" and aggregate models. The impact of this difference could be lessened by "re-aggregating" the detailed model and "reapplying" the requirements. All that may be required is a check that the models are consistent at the aggregate level. If there are major differences, however, the aggregated model should be reconstructed and the requirements reapplied.

5.2 Directions

The goal of Task 3 is to define the AEGIS maintenance methodology. The application of the requirements to the aggregate model presented above may not seem a good presentation of the "ideal" model. However, a good foundation is laid for the analysis required for Task 3.

What the "ideal" model provides is a correlation between the existing process and the requirements for the methodology. This relationship can be utilized to identify the "delta" between the current maintenance process and the one implied by the requirements. Direction for defining the methodology is provided by this "delta" which indicates the need for improvement.

Because of this work, Task 3, can begin with an identification of the activities which are in need of improvement. Further, potential points of improvement are marked. Task 3 can consider alternative methods for meeting requirements and the opportunities afforded by different points of application. The result should lead to perceived improvements in effectiveness and efficiency provided that they are accepted by the AEGIS community.

REFERENCES

[BUNG90] Bundy, Gary (1990), Personal Communication

[FAUS89] Faulk, S.R. and D.L. Parnas (1989), "On Synchronization in Hard-Real-Time Systems," *Communications ACM 31* 3, 1989, 274-287.

[GAMS88] Gamalel-Din, S.A. and L.J. Osterweil (1988), "New Perspectives on Software Maintenance Processes," In *Proceedings of the Conference on Software Maintenance - 1988*, IEEE Computer Society Press, 14-22.

[NANR88] Nance, R.E. and J.D. Arthur (1988), "The Methodology Roles in the Realizaiton of a Model Development Environment," In *Proceedings of the 1988 Winter Simulation Conference*, M.A. Abrams, P.L. Haigh, and J.C. Comfort, Eds. IEEE, Piscataway, NJ, 220-225.

[NANR89] Nance, R. E., B. J. Keller, and D. Boldery (1989), "Documentation Production Under Next Generation Technologies," Technical Report SRC-89-001, Systems Research Center, Virginia Tech, Blacksburg, VA.

[NANR90] Nance, R. E., J. D. Arthur, and B. J. Keller (1990), "Requirements for a Software Maintenance Methodology," Technical Report SRC-90-001, Systems Research Center, Virginia Tech, Blacksburg, VA.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188
1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Systems Research Center SRC-90-006	5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Systems Research Center	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Surface Warfare Center
6c. ADDRESS (City, State, and ZIP Code) 320 Femoyer Hall, Virginia Tech, Blacksburg, Virginia 24061-0251		7b. ADDRESS (City, State, and ZIP Code) Dahlgren, Virginia 22448-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Surface Warfare Center	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Dahlgren, Virginia 22448-5000		10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Development of an AEGIS Maintenance Methodology, Task 2: Models of the AEGIS Maintenance Process, Interim Status Repo

12. PERSONAL AUTHOR(S)
Benjamin Keller, Richard E. Nance, and James D. Arthur

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim Status	FROM 12/15/89 TO 7/15/90	August 1990	50

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Software maintenance, methodology, principles, maintenance models, methodology requirements

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Covering the Task 2 effort in the development of an AEGIS software maintenance methodology, the derivation of requirements from maintenance principles is summarized. Three models of the maintenance process provide the basis for identifying points of application. The mid-level (aggregate) model is the major focus of the methodology definition. Recommendations address the potential uses of different modeling perspectives, better balance between product and process assessment, use of metrics, and the selection of maintenance tools.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
[✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE