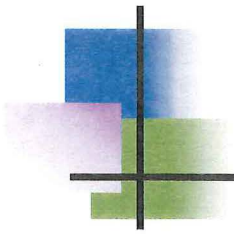


Systems Engineering Project

Test Procedure Validation Process

Prepared by:
Tony Hernandez
12/11/2010

Executive Summary

Objective

- Standardize a Satellite Program's Test Procedure Validation Process.
  - Align it to the organization's existing processes

Justification

- Reduction in the number of validation process issues (rework) currently plaguing Program X.

Method

- Utilize techniques obtained from Systems Engineering course work.

Executive Summary (cont.)

- Below is a table showing the top three causes of satellite test procedure development rework.

Development Process issues	**Validation Process Issues**	**Configuration Management Issues**
Poor Requirement gathering, Bad or Missing Requirements	Process is incomplete and is either not efficient or does not capture problems. Need better tools. Peer reviews not performed, etc.	Files not controlled, modified on the fly, multiple users editing the same files. None or Poor Configuration Management Software

Introduction

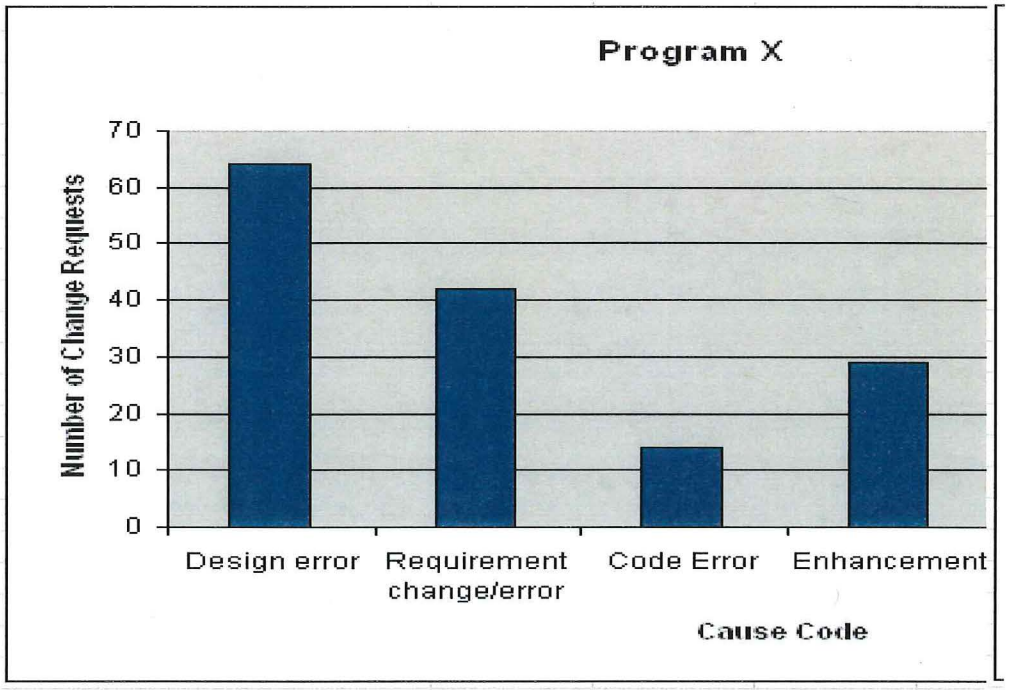
- System Test is an organization that in responsible for testing a product at various levels of integration. The "product" in this case will be a Satellite.
- Test Procedures are developed and utilized for testing in System Test.
- Test procedures are developed via a Test Procedure Development Lab (TP Dev).
- Test procedures are validated against a Satellite Simulator. This is called the "validation process".
- Disconnects in the validation process or lack of, affect the final test procedure product.
- Program X has a lot of rework due to a poor validation process.

Introduction (cont.)

- Program X is currently in a Re-occurring phase which means that all their development is complete.
- Due to a deficient initial validation process, the final released product is of poor quality and usually results in further rework.
- In a re-occurring test phase, rework (changes) are done through a Change Process.
- Validation occurs within the Change Process.

Top issues on Program X

- "Design Errors" in Test Procedure development is the largest source of issues.
  - These types of errors should be caught during the validation process.
  - A review of the Design Error issues points to a poor validation process as the root cause for these issues.



Risk Analysis

There are many risks to not having a validation process or having a poor validation process which can lead to:

- A large number of test failures during satellite test
- Possible damage of flight hardware and/or harm to test personnel.
- Impacts to Cost and Schedule.
- Potential loss of the contract and/or future business

Benefits

An improved validation plan will be lean and efficient, resulting in a quality product.

- Reduced number of escapes
- Reduced need for rework
- Reduced downtime
- Reduced risk to flight hardware and/or personnel
- Standardized tools and processes
- Sharing of resources
- Sharing of Lessons Learned
- Reduced training costs

Requirements

The Test Procedure Development process includes validation.

This process is guided by Enterprise Processes/Requirements. The following items are Enterprise Processes which flow down as requirements to the System Test Organization:

- Document processes and procedures
  - All processes and procedures need to be documented and released.
- Perform Requirements Development
  - Ensures that requirement gathering, analysis and documentation is performed.
- Perform Verification
  - Verifies the procedure is performing to the requirements.
- Perform Validation
  - Ensures the procedures do what they are suppose to do.

Requirements (cont.)

- Perform Configuration Management
  - All documents and software used needs to be properly controlled. This includes having the proper signatures where applicable. Configuration management is a quality control tool.
- Perform Risk Management
  - Thorough identification and review of risk must be performed.

Assumptions

Goal is to reduce the number of "Design Errors" on Program X by improving the validation process. To do this some assumptions need to be made:

- Obtained approval to make hardware configuration changes to the existing validation flow. Moving some hardware around will enable flow and improve capabilities.
- Approved to make updates to the validation tools. The ability to update these tools will improve the success of the validation process.
- Approved to change Configuration Management flow.

Process Flow Comparison

Before making any changes to Program X's current process, we need to draw a flow chart of the two processes to be compared and perform a top level comparison.

1) Draw Program X's Change Process.
   - Refer to the backup slides Figure 1.

2) Draw System Test Change Process.
   - Refer to the backup slides Figure 2.

Process Flow Comparison (cont.)

3) Compare the two processes at a "Top Level".

- When looking over the flows, a major area for improvement is identified. By relocating Program X's development server, several advantages are quickly apparent:

  - Improved flow; approximately 15 steps removed from flow
  - Reduced movement of people and data
  - Improved accessibility
  - Enables facilitated validation
  - Enables use of common tools such as Configuration Management tools.

  However, there may still be non-value added steps in this process which still make it in efficient.

Lean Engineering

Simply comparing the two flows side by side is not enough. Once we have drawn flow charts of the current flows for Program X and System Test we need to develop the "Future State" which is the desired flow.

In order to develop the "Future State", we need to utilize some Lean Engineering tools to make an efficient new flow for Program X.

- Lean has Five basic principles:
  - Specify Value
  - Identify the Value Stream
  - Make value flow continuously
  - Let customers pull value
  - Pursue perfection

  *Respect People

The next several slides break down each of these principles in more detail. They are utilized in developing a "Current" and "Future" state map of the TP Dev Process.

VSM

- The "Current State" was captured in the Top Level Process Comparison slides. The following method breaks down that process even further to identify every step since some steps may be missing from the current documented process.

- A technique used in Value Stream Mapping involves posting the flow with post-its onto the wall.

- Below is an example of Program X's change process laid out with post-its so that the stickies could be moved around easily and various flows examined.

VSM (cont.)

- The Current State flow was laid-out by "walking" through the Change Process end to end and identifying any undocumented steps.
- Each sticky was labeled with the words VA, NVA, R-NVA, which stood for Value Added, Non-Value Added and Required Non-Value Added.
- Since Program X had slightly different steps than System Test, we used the System Test flow as a guideline and moved the Program X post-its around to try to standardize Program X's process to the System Test process along with removing any waste that may have existed in Program X's process.

VSM (cont.)

In developing the Future State Map, we look to eliminate the NVA steps that exist in the Current State. The Lean process talks about "Ohno's Seven Types of Waste". The seven types of waste are:

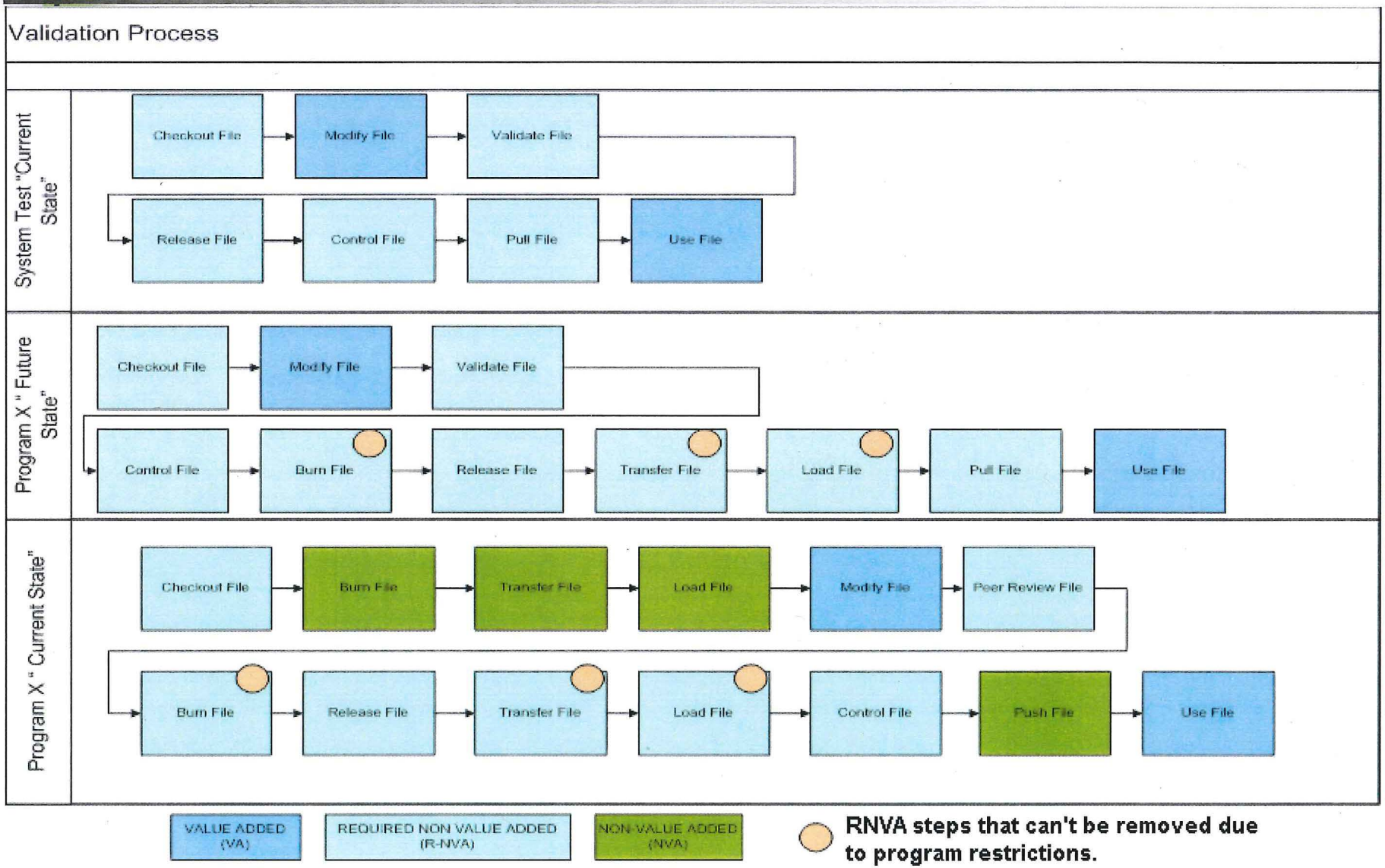
1. over-production
2. inventory
3. transportation
4. unnecessary movement
5. waiting
6. defective outputs
7. over-processing

VSM (cont.)

- Items 4, 5, 7 (unneccesary movement, waiting and defective outputs) are the items that are applicable to this project.
  - People were moving files from one location to another manually by hand carrying a CD. We were able to address one instance of this in our re-architecture. The remaining "hand carry" item could not be eliminated due to program restrictions.
  - Waiting occurs in this process when the Test Engineer is waiting for the file to be transferred and for the Test Engineer who is waiting to use the file. We reduced these wait times by removing non-value added steps which in turn reduces the cycle time.

The next slide shows the final product, the Future State side by side with the System Test Current State process map.

VSM (cont.)


Validation Process
System Test "Current State"
Checkout File
Modify File
Validate File
Release File
Control File
Pull File
Use File
Program X " Future State"
Checkout File
Modify File
Validate File
Control File
Burn File
Release File
Transfer File
Load File
Pull File
Use File
Program X " Current State"
Checkout File
Burn File
Transfer File
Load File
Modify File
Peer Review File
Burn File
Release File
Transfer File
Load File
Control File
Push File
Use File
VALUE ADDED (VA)
REQUIRED NON VALUE ADDED (R-NVA)
NON-VALUE ADDED (NVA)
RNVA steps that can't be removed due to program restrictions.

VSM (cont.)

- On the previous slide, you can see that the Future State map takes into account basic lean principles 3, 4, 5 and 6 mentioned previously, which are: 3) Make value flow continuously, 4) Let customers pull value, 5) Pursue perfection, 6) Respect People.

- The Future State enables flow where possible and adds "pulls" where there use to be "pushes". The Future State process will be monitored and continuously improved as opportunities are presented. The whole process takes into account the people performing these steps.

VSM (cont.)

- The table below shows the results of the VSM. By performing the VSM, we were able to obtain reductions in Cycle-Time, Process Steps, Number of Handoffs and Travel Distance.
  - Refer to Table 1 in backup slides for raw data.

Category	Reduction
Cycle-Time	17%
Process Steps	31%
No. of Handoffs	33%
Travel Distance	44%

TP Dev Process Document Updates

- Program X's Change Process flow has been optimized and aligned with System Test using Lean Engineering tools.
- We then reviewed any additional process documentation and made the necessary updates so that it supports the optimized flow.
- A process matrix was developed to identify Program X's process escapes. The matrix shows the documented processes used by System Test and Program X for TP Development. See next slide for matrix.

TP Dev Process Matrix (existing)

Documented Proceseses				
Process	**Subset**	**Enterprise Process (Top Level)**	**System Test Process**	**Program X Process**
TP Dev Training	N/A	ENT-3376, "Test Planning & Documentation"	Yes	No
TP Dev Process	TP Dev Process	ENT-3376, "Test Planning & Documentation"	SYST 300-08-2078, "System Test Procedure Development Process"	No
Validation/Verification Process	N/A	ENT-1916, "Verify and Validate Products/Services", SYST 298-07-040, "Perform Peer Reviews Method"	SYST 300-08-983 "Test Development, Validation and Change process".	No
Review Process	Review Process	SYST 298-07-040, "Perform Peer Reviews Method"	SYST 298-07-040, "Perform Peer Reviews Method"	No
Configuration Management Process	N/A	ENT-8621 "Configuration Management Requirements and Objectives", ORG-9642, "Manage Configurations and Data"	ORG-9642, "Manage Configurations and Data"	No
Approval and Release Process	Approval Process	ENT-9676 Engineering Approval & Release	SYST 300-09-00136, "Engineering Approval and Release"	PRGX-0379; PRGX-0686
TP Change Process	CCB Command Media	ORG-1715 Engineering Change Control Process	ORG-0815 Test Engineering Change Control Process (TE-ECR)	ORG-0815 Test Engineering Change Control Process (TE-ECR)
Database Change Process	Database Change Process	SYST 300-08-1948 "Database Change Request Review Process"	System Test Database Change and Delivery Process SYST 300-09-00566	No
Special Test Process (STR)	STR Process	ENT-4376 "Special Test Requests (STRs)	ENT-4376 and SYST 300-08-280, "STR – Special Test Request" FORM	ENT-4376 and SYST 300-08-280, "STR – Special Test Request" FORM

TP Dev Process Document Updates (cont.)

- The process matrix shows that processes are missing for Program X.
- System Test and Program X process documentation is not aligned.
- Some process documentation is too detailed. Lots of steps not applicable to Program X.
- Identified documents to update and generalize to make common.

TP Dev Process Document Updates (cont.)

Two documents were identified to be updated. These were the TP Dev Process document (SYST 300-08-2078) and the Validation Process (SYST 300-08-983).

- The team reviewed the documents and they were redlined.
- The redlines were reviewed with all Stakeholders.
- The redlines were approved and incorporated.

The Revised documents now align with Program X's new flow and the System Test baseline flow.

TP Dev Process Document Updates (cont.)

- The Process Matrix was updated after document revisions.
- The updated table shows that Program X is now in line with the System Test processes.
- Program X had one process that was specific to their program so they continue to use that process. See "blue" item in matrix on next slide.

TP Dev Process Matrix (future)

Documented Proceseses				
Process	**Subset**	**Enterprise Process (Top Level)**	**TP Dev Lab Process**	**Program X Process**
TP Dev Training	N/A	ENT-3376, "Test Planning & Documentation"	Yes	Yes
TP Dev Process	TP Dev Process	ENT-3376, "Test Planning & Documentation"	SYST 300-08-2078, "System Test Procedure Development Process"	SYST 300-08-2078, "System Test Procedure Development Process"
Validation/Verification Process	N/A	ENT-1916, "Verify and Validate Products/Services", SYST 298-07-040, "Perform Peer Reviews Method"	SYST 300-08-983 "Test Development, Validation and Change process".	SYST 300-08-983 "Test Development, Validation and Change process".
Review Process	Review Process	SYST 298-07-040, "Perform Peer Reviews Method"	SYST 298-07-040, "Perform Peer Reviews Method"	SYST 298-07-040, "Perform Peer Reviews Method"
Configuration Management Process	N/A	ENT-8621 "Configuration Management Requirements and Objectives", ORG-9642, "Manage Configurations and Data"	ORG-9642, "Manage Configurations and Data"	ORG-9642, "Manage Configurations and Data"
Approval and Release Process	Approval Process	ENT-9676 Engineering Approval & Release	SYST 300-09-00136, "Engineering Approval and Release"	PRGX-0379; PRGX-0686
TP Change Process	CCB Command Media	ORG-1715 Engineering Change Control Process	ORG-0815 Test Engineering Change Control Process (TE-ECR)	ORG-0815 Test Engineering Change Control Process (TE-ECR)
Database Change Process	Database Change Process	SYST 300-08-1948 "Database Change Request Review Process"	System Test Database Change and Delivery Process SYST 300-09-00566	System Test Database Change and Delivery Process SYST 300-09-00566
Special Test Process (STR)	STR Process	ENT-4376 "Special Test Requests (STRs)	ENT-4376 and SYST 300-08-280, "STR – Special Test Request" FORM	ENT-4376 and SYST 300-08-280, "STR – Special Test Request" FORM

Validation Tools

- Test procedure development and validation depends on quality tools.
- The right tools can minimize down time, failures, safety issues, schedule delays, cost uppers, etc.

Validation Tools

A Satellite Simulator is a tool available for use during Test Procedure Development.

- Current State
  - Seldom used by Program X due to inconvenience.
  - Simulator on a different network than the TP Development.
- Future State
  - Development Lab on same network as the simulator.
  - Simulator used for all validations
  - The simulator scenario files were customized for ground testing which improved accuracy of simulator.

Trade-Offs in validation methods

- There are two validation methods to compare:

1) Single versus Multiple Test Procedure validations
2) Simulator versus a Compiler

Trade-Offs in validation methods

Single versus Multiple Test Procedure validations

- Single Validations
  - Ideal
  - TPs get released faster.
  - No Peer reviews
  - Human error – increased escapes
  - Increased need for rework
- Multiple Validations
  - Peer reviews
  - Reduces number of escapes
  - Higher quality product
  - Slows process (upfront)

Trade-Offs in validation methods

Compiler versus Simulator

- Compilers (aka syntax checkers)
  - Simply "compiles" the test scripts
  - Identifies code syntax errors which cause compiler to abort.
  - Tells operator that there are no syntax errors.
  - Does not validate code logic.
  - Does not validate commands
  - Does not validate telemetry checks.
- Simulators (hardware and software)
  - Emulate the functionality of the spacecraft.
  - Validate all commands and telemetry checks
  - Validate syntax and code logic.
  - Comprehensive
  - Most "realistic"

Conclusion/Lessons Learned

This project was triggered by a large amount of rework occurring on a satellite program (Program X). The following was accomplished:

- Determine root cause of problem.
- Standardized the validation process between Program X and System Test.
- Utilized Lean Engineering to develop Value Stream Maps of the current and future states of the Program X process.
- Made physical configuration changes to enable flow.
- Documents updated, reviewed by stakeholders, and released.
- Tools improved and made common.
- Team trained in new process.

Figure 1: Program X current process

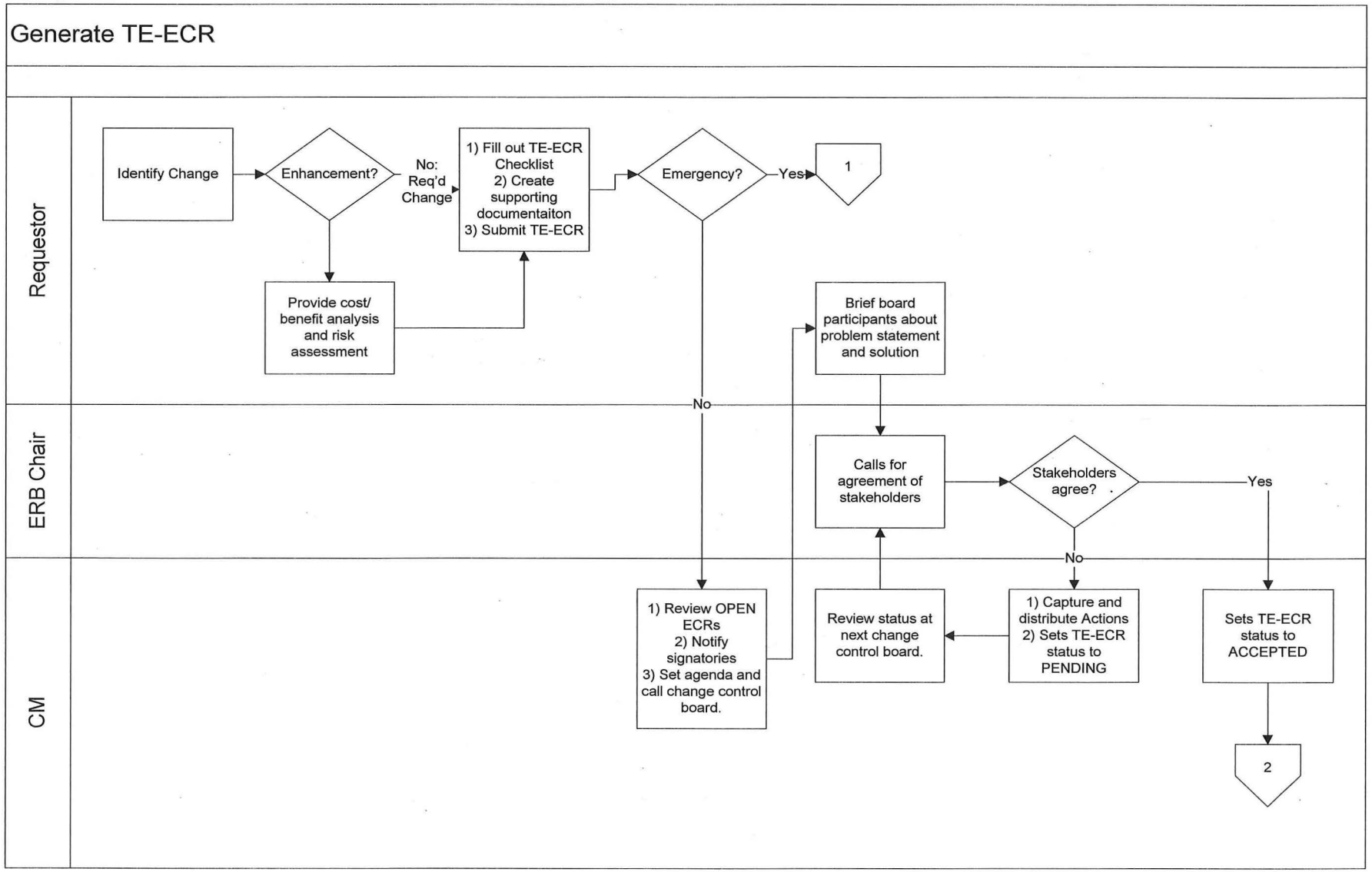


Figure 1: Program X current process (cont.)

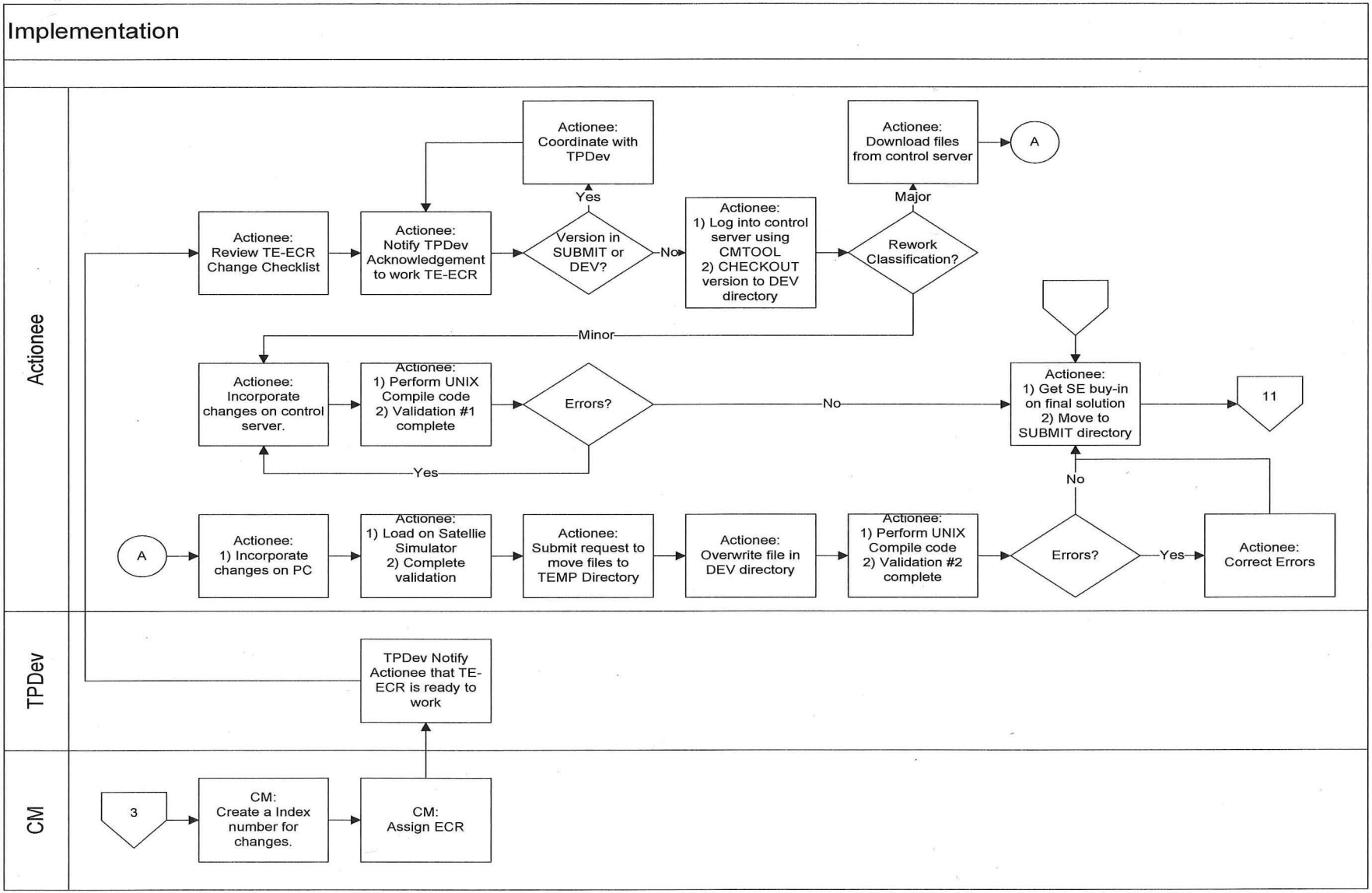


Figure 1: Program X current process (cont.)

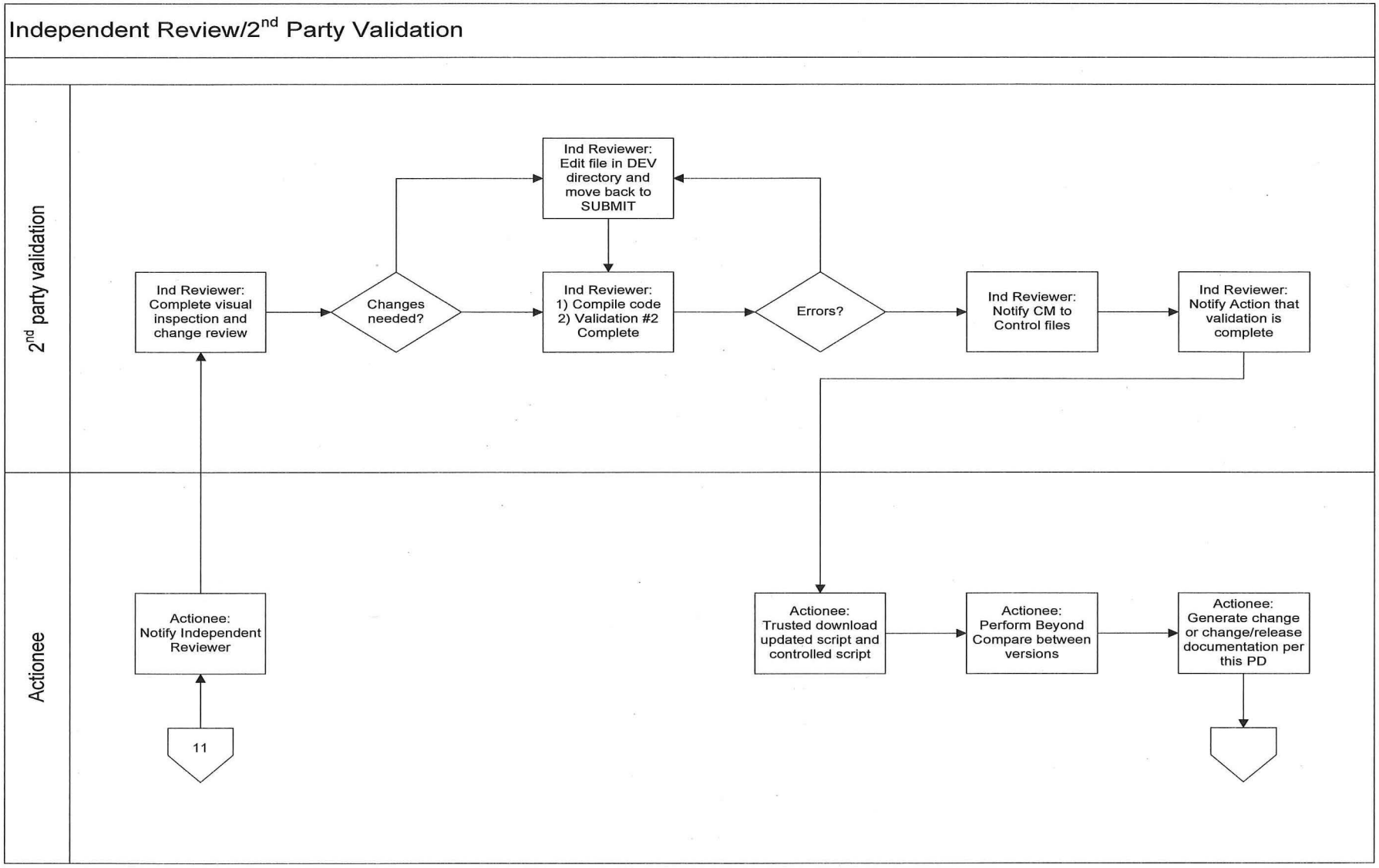


Figure 1: Program X current process (cont.)

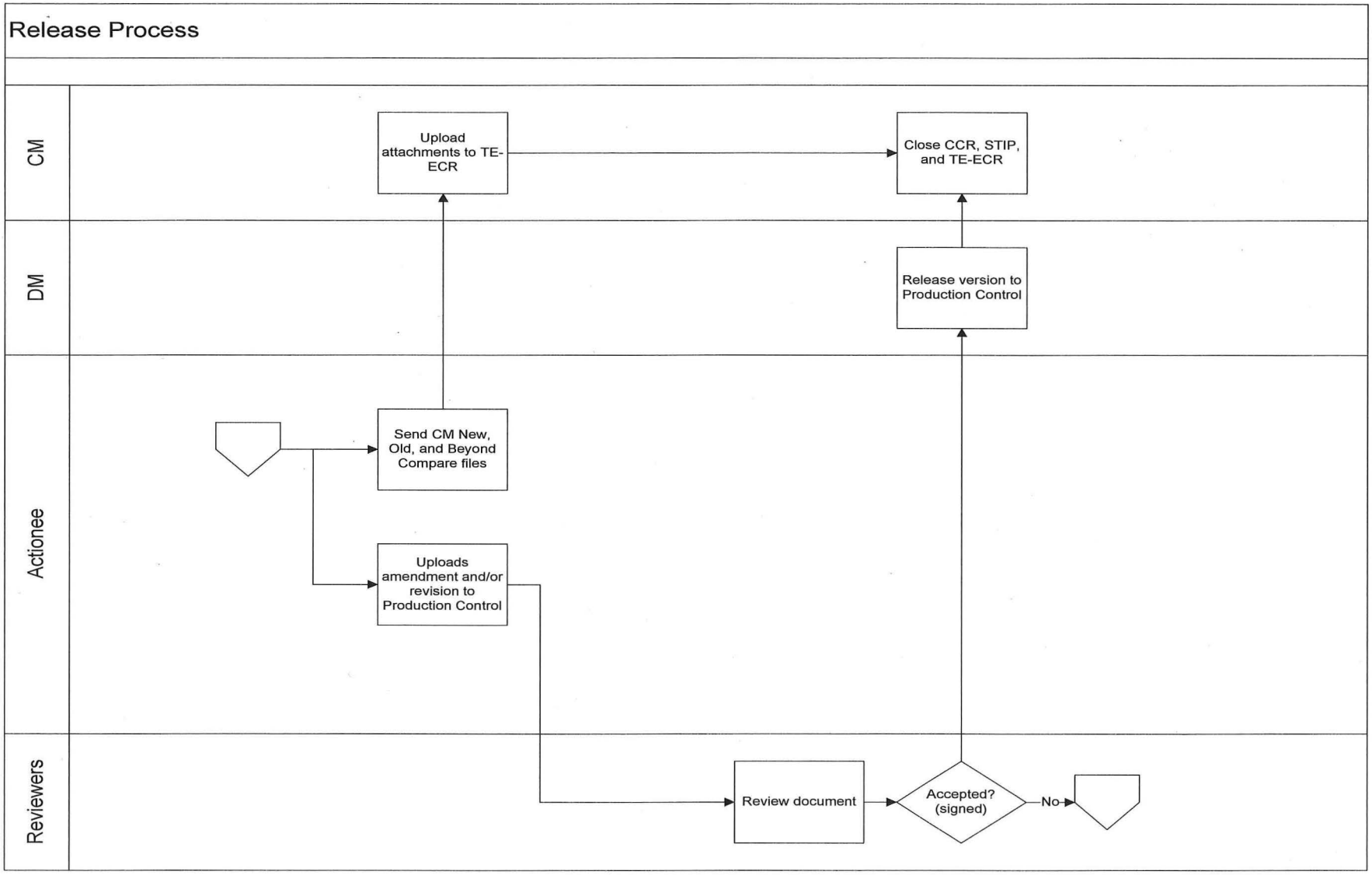


Figure 2: System Test current process (cont.)

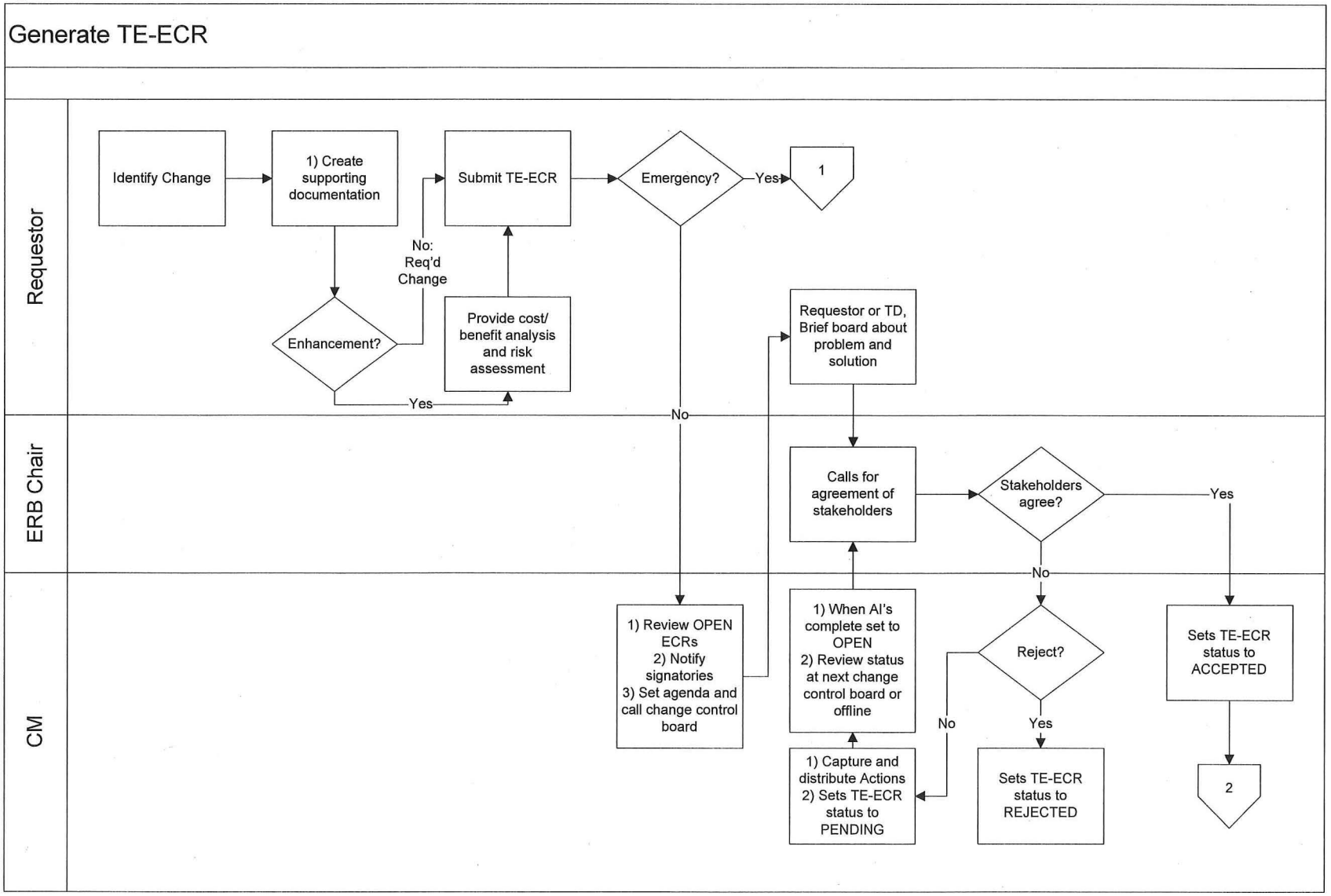


Figure 2: System Test current process (cont.)

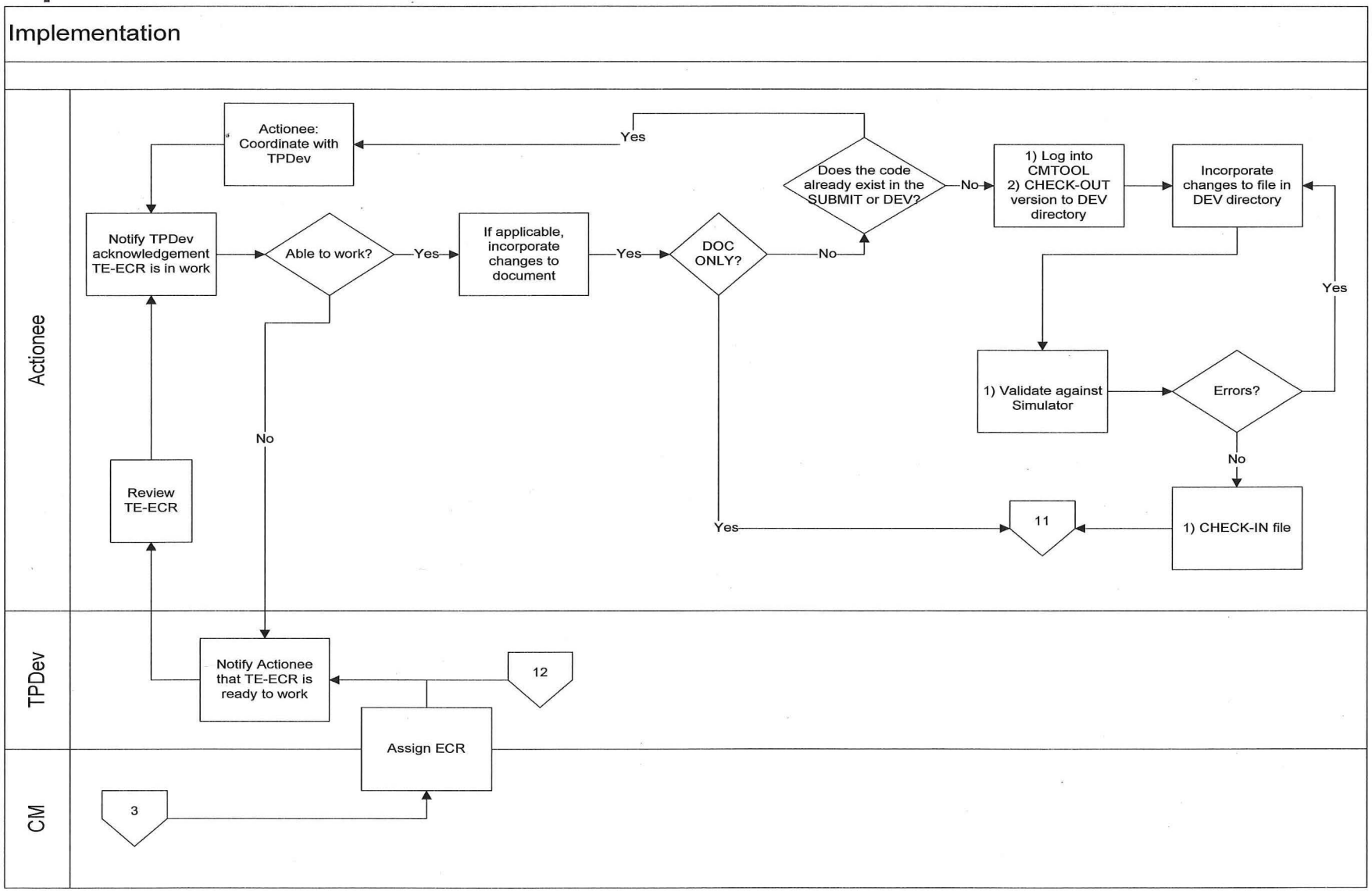


Figure 2: System Test current process (cont.)

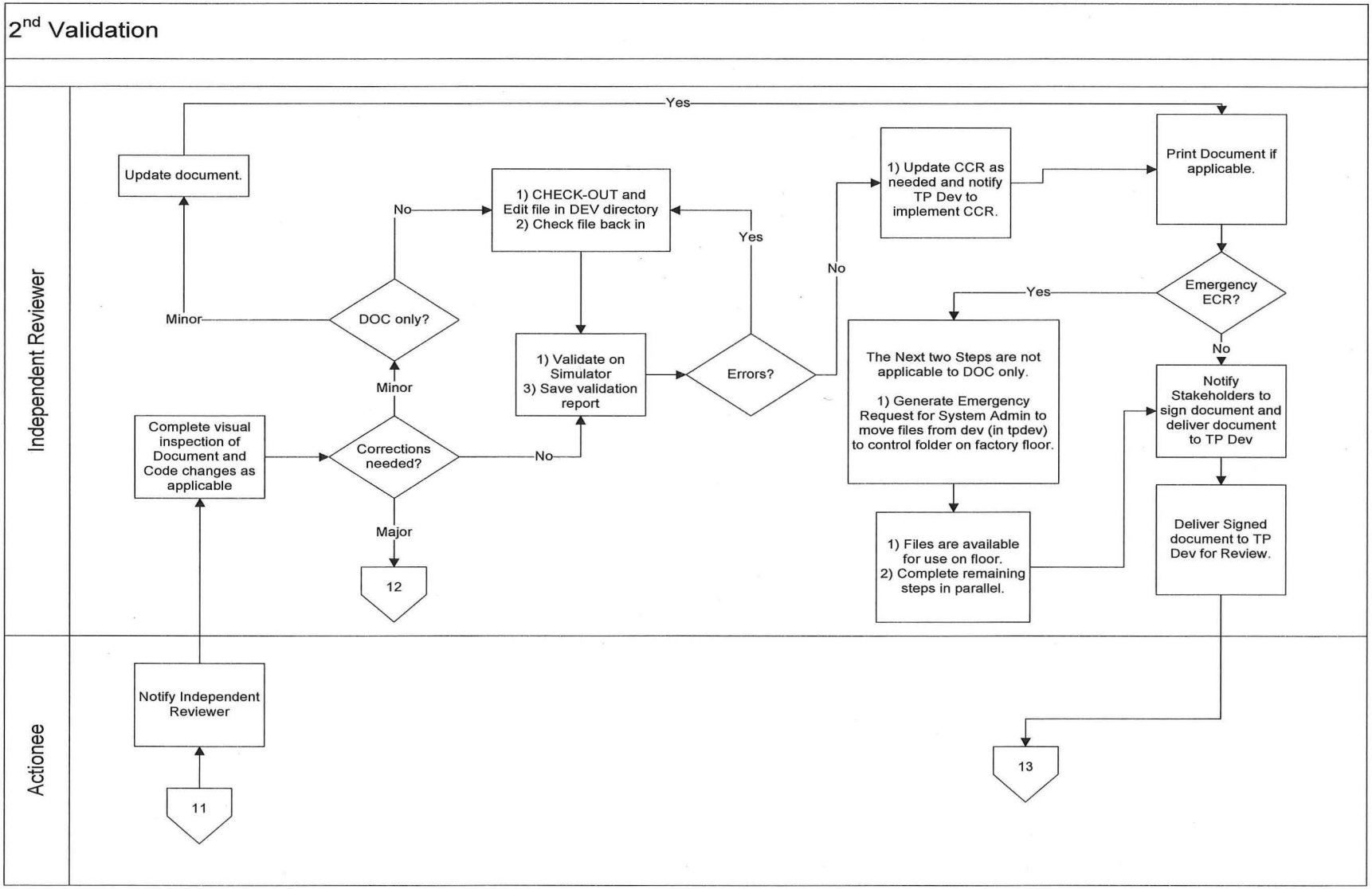


Figure 2: System Test current process (cont.)

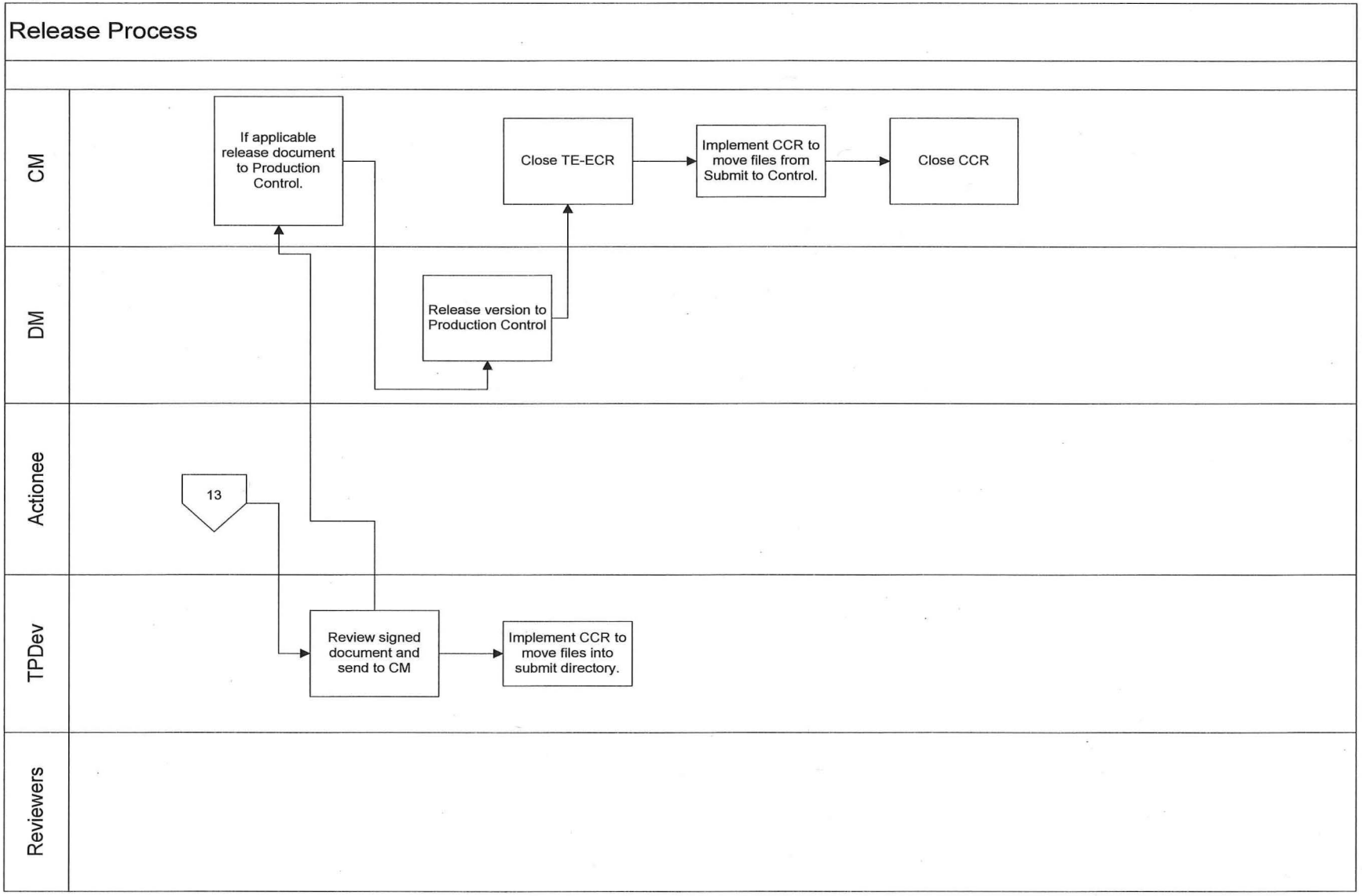


Table 1: Raw Data

	Program X Current State	Program X Future State	System Test Current State
Checkout File	5	5	5
Burn File	10	0	0
Transfer File	15	0	0
Load File	5	0	0
Modify File	30	30	30
Peer Review/Validate File	30	30	30
Burn File	10	10	0
Release File	30	30	30
Transfer File	15	15	0
Load File	5	5	0
Control File	20	20	20
Push/Pull File	5	5	5
Use File	0	0	0
Cycle Time Total (minutes)	180	150	120
Reduction	17%		

	Program X Current State	Program X Future State	System Test Current State
Checkout File	1	1	1
Burn File	1	0	0
Transfer File	1	0	0
Load File	1	0	0
Modify File	1	1	1
Peer Review/Validate File	1	1	1
Burn File	1	1	0
Release File	1	1	1
Transfer File	1	1	0
Load File	1	0	0
Control File	1	1	1
Push/Pull File	1	1	1
Use File	1	1	1
Process Steps	13	9	7
Reduction	31%		

Table 1: Raw Data (cont.)

	Program X Current State	Program X Future State	System Test Current State
Checkout File	TE	TE	TE
Bum File	TE		
Transfer File	TE		
Load File	CM		
Modify File	TE	TE	TE
Peer Review/Validate File	TE2	TE2	TE2
Bum File	TE	CM	
Release File	CM	CM	CM
Transfer File	CM	CM	
Load File	CM	CM	
Control File	CM	CM	CM
Push/Pull File	CM	TE	TE
Use File	TE	TE	TE
Number of Handoffs (total)	6	4	4
Reduction	33%		

	Program X Current State	Program X Future State	System Test Current State
Checkout File	0	0	0
Bum File	0	0	0
Transfer File	400	0	0
Load File	0	0	0
Modify File	0	0	0
Peer Review/Validate File	0	0	0
Bum File	0	0	0
Release File	100	100	100
Transfer File	400	400	0
Load File	0	0	0
Control File	0	0	0
Push/Pull File	0	0	0
Use File	0	0	0
Travel Distance (feet)	900	500	100
Reduction	44%		