

# General Information, Eligibility, and Entitlement Manual

## Chapter 7 - Contract Administrative Requirements

---

### Table of Contents (Rev. 6, 05-28-04)

- 10 – Contracting and Subcontracting – Not yet available
- 20 – Publishing by Contractors and Restrictions on Promotional Activity – Not yet available
- 30 – Files Maintenance – Not yet available
- [40 – Shared System Maintainer and Medicare Contractor Responsibilities for System Releases](#)
- 40.1 – Claims Processing—Not yet available
- 40.2 – *Release Software*
- [40.3 – Shared System Testing Requirements for Maintainers, Beta Testers, and Contractors](#)
- [40.3.1 – Maintainers and Beta Testers – Required Levels of Testing](#)
- [40.3.2 – Minimum Testing Standards for Maintainers and Beta Testers](#)
- [40.3.3 – Testing Standards Applicable to all Beta Testers](#)
- [40.3.4 – Testing Requirements Applicable to the CWF Beta Tester](#)
- [40.3.5 – Contractor \(User\) Testing Requirements](#)
- [40.3.6 – Testing Requirements Applicable to all CWF Data Centers \(Hosts\)](#)
- [40.3.7 – Timeframe Requirements for all Testing Entities](#)
- [40.3.8 – Testing Documentation Requirements](#)
- [40.3.9 – Definitions](#)
- [40.3.10 - Test Case Specification Standard](#)

## **40 – Shared System Maintainer and Medicare Contractor Responsibilities for System Releases**

*A3-3600ff., A3-3800ff., B3-4000ff., B3-6000ff., (COB SOW - §4.8)*

*(Rev. 6, 05-28-04)*

### **40.1 – Not yet available**

### **40.2 – Release Software**

*(Rev. 5, 05-07-04)*

*CMS intends to continue to closely manage standard system software changes to assure that an effective change control process is in place. This means that maintainers must receive approval from their CMS system maintenance lead (see section VI) or CMS project officer before any follow-up release by the standard maintainer can be scheduled and installed.*

#### **Control of System Changes**

*All maintainers of the standard systems (CWF, FISS, APASS, MCS, VMS, GTEMS, and HPBSS systems) must use the same quarterly release schedule, i.e., on or about January 1, April 1, July 1, and October 1. The specific schedule for each quarterly release will be determined by CMS.*

*All follow-up release changes (except emergencies) to the quarterly schedule must be held and released on a predetermined schedule in coordination with CMS. Emergency changes may be released as problems are identified without prior approval. The schedule for follow-up release of changes must be forwarded to your CMS system maintenance lead or CMS project officer for prior approval.*

*Follow-up release changes are to be limited to the correction of priority 1 and 2 problems and errors that prevent effective operation of the production system. Priority 3, priority 4 and/or priority 5 problems may be corrected in a follow-up release when pre-approved by CMS. The CMS maintenance lead will advise you of the approval decision within 24 – 48 hours.*

*If a system problem is identified, Medicare organizations must submit documentation to their CMS system maintenance lead outlining the problem and the reason correction is needed at this time. Section V of this instruction outlines the minimum information required by CMS for approval.*

#### **Problem Priority Classifications for Follow-Up Releases**

*Listed below are CMS's problem priority classifications and examples. These are similar to the problem priority classifications that were used for the Y2K re-certification testing period.*

##### **Priority 1 Classification**

*Production:*

*The problem prevents the accomplishment of a mission critical capability for which no acceptable workaround is known.\**

*This priority also includes problems where code must be fixed immediately in order for the normal production region functions or services to continue. For example, if the production region is down in a job resulting in an incomplete cycle or the system is pricing a significant volume of claims incorrectly causing over or under payment. The maintainer may make priority 1 changes on its own authority. These corrections must be reported to the CMS maintenance lead or to the project officer the next business day.*

*Examples:*

- *ABENDS on-line or batch (Inability to run a cycle)*
- *Inaccurate payment or no payment of claims (significant impact/high volume)*
- *Necessary file updates cannot be accomplished (payment files, history files)*
- *Interface failures affecting claims processing*

*Beta/User Acceptance Testing:*

*The problem would prevent the accomplishment of a mission critical capability if the current test software is moved into the production environment. This priority also includes problems where code must be fixed immediately in order for the normal test region functions or services to continue. For example, if the test region is down in a job causing the cycle to not complete or the system is pricing claims incorrectly with a potentially significant claim volume or payment impact, the issue would be classified as a priority 1. The maintainer must work immediately to code a fix to be installed before moving the software into production.*

*Examples:*

- *ABENDS; inability to run a cycle or test*
- *Inaccurate payment or no payment of claims (potentially significant impact)*
- *Necessary file updates cannot be accomplished (payment files, history files)*
- *Interface failures affecting test conditions*

## **Priority 2 Classification**

*Production:*

*The problem adversely affects the accomplishment of a mission critical capability so as to degrade performance and for which no acceptable work-around is known.\* This means the problem adversely affects the payment of benefits with a small claim volume or payment impact, the completion of CMS required reporting, or inaccurate information is being sent providers, beneficiaries or CMS. For example, if the information on an outgoing document to the provider community or Medicare Summary Notice is incorrect, the issue would be classified as a priority 2. The system maintainer must work with the CMS maintenance lead for approval to implement a fix.*

*Examples:*

- *Inaccurate payment or no payment of claims (small impact/low volume)*
- *Inaccurate CMS required report*
- *Inaccurate messages to the beneficiary, provider or CMS*
- *ABENDs with limited impact (ex. One contractor)*

*Beta/User Acceptance Testing:*

*The problem would adversely affect the accomplishment of a mission critical capability so as to degrade performance if current test software is moved into the production environment. This means the problem adversely affects the payment of benefits with a potentially small claim volume or payment impact, the completion of CMS required reporting, or inaccurate information is being sent to providers, beneficiaries or CMS. For example, if the information on an outgoing document to the provider community is incorrect, the issue would be classified as a priority 2. The maintainer must work immediately to code a fix to be installed before moving the software into production.*

*Examples:*

- *Inaccurate payment or no payment of claims (potentially small impact)*
- *Inaccurate CMS required report*
- *Inaccurate messages to the beneficiary, provider or CMS*

### ***Priority 3 Classification***

*Production:*

*The problem adversely affects the accomplishment of mission critical capability so as to degrade performance and for which an acceptable workaround is known.\**

*This means the problem could have significant impact but the work-around alleviates the impact. This allows the system maintainer adequate time to code a fix and sufficiently test before the corrected software is delivered for production installation. The system maintainer must work with the CMS maintenance lead to implement a fix.*

*Examples:*

- *Impact of problem could be significant or minimal*
- *Problem correctable by contractor workaround\**
- *ABENDs with an acceptable workaround\**

*Beta/User Acceptance Testing:*

*The problem would adversely impact the accomplishment of a mission critical capability so as to degrade performance if current test software is moved into the production environment.*

*If moved into the production environment before correcting an acceptable workaround could be instituted to prevent the adverse impact.\*\* The system maintainer must work immediately to code a fix to be installed before moving the software into production.*

*Examples:*

- *Potential impact of problem could be significant or minimal*
- *Problem affects CMS required reporting*

#### **Priority 4 Classification**

*Production:*

*The problem is an operator inconvenience or annoyance, which does not affect a required mission essential capability. The system maintainer must request approval to code and implement a fix from its CMS maintenance lead.*

*Examples:*

- *Problems affects non-mission critical functions*
- *Operational procedure with workload impact that should be automated*
- *Impact of problem is minimal*
- *Correctable by contractor workaround\**

*Beta/User Acceptance Testing:*

*The problem is a test inconvenience or annoyance, which does not affect a required mission essential or test capability. If moved into the production environment before correcting, an acceptable workaround could be instituted to prevent the inconvenience.\*\* The system maintainer should work immediately to code a fix to be installed before moving the software into production.*

*Examples:*

- *Problem affects non-mission critical functions*
- *Operational procedure with workload impact that should be automated*
- *Impact of problem is minimal*
- *Correctable by contractor workaround\**

#### **Priority 5 Classification**

*Production:*

*All other documented system problems. These could include operator errors, an inability to reproduce the reported problem, a problem with insufficient information, or documentation errors. The system maintainer should request approval from the CMS maintenance lead before coding and implementing any system enhancements.*

*Examples:*

- *Contractor requested enhancements*
- *Documentation errors (i.e. Business requirements)*
- *Problem affects non-mission critical functions*
- *Minimal impact*

*Beta/User Acceptance Testing:*

*All other documented system test problems. These could include operator errors, an inability to reproduce the reported problem, a problem with insufficient information, or test documentation errors. The system maintainer should work to correct these issues as soon as possible but any system enhancements should be discussed with the CMS maintenance lead.*

*Examples:*

- *Test region or processing enhancements*
- *Test documentation errors (i.e. business requirements)*
- *Problem affects non-mission critical test functions*
- *Minimal impact*

*\* An acceptable workaround is a temporary alternative solution to a confirmed problem in the shared system that will insure the contractor is able to accomplish a mission critical capability. What makes the workaround “acceptable” is it must be agreeable to both the maintainer and contractor and does not cause an excessive burden to the contractor. If the maintainer and contractor cannot come to an agreement on what is “acceptable” the decision will be made by CMS.*

*\*\* CMS does not recommend using workarounds in the test region in order to “pass” test cases. The institution of a workaround should be used in order to implement a CMS mandate where the system maintainer may not have time to adequately code a fix before the software is delivered for production installation.*

***Routine File Maintenance/Updates***

*CMS does not require pre-approval or special documentation of routine file maintenance/updates or other routine activities necessary for effective operation of the Medicare system, Medicare processes and/or testing (e.g., MR/UR screen updates, provider and beneficiary file updates). All contractors and data centers should continue with their normal file maintenance routines.*

***Testing Prior to Installation of CMS Approved Follow-up Releases***

*CMS explains expectation for each Medicare organization’s testing responsibility (i.e., standard system maintainer testing, contractor testing, CWF host testing, Beta testing).*

***Information Required for Requesting CMS Approval***

*The following must be submitted to the CMS maintenance lead or project officer when requesting*

*that a problem be implemented in a follow-up release. If the system maintainer already has a process in place for communicating system problems to CMS, that process may be used as long as all information below, at a minimum, is captured.*

**MAINTAINER NAME:**

*Problem Description:*

*Brief non-technical business description of the fix.*

*How Found:*

*Explain how the problem was found. Also explain why you believe it was not found by release testing.*

*Problem Impact:*

*This information is needed to determine the scope of the problem in terms of payments, provider types, beneficiaries, number of potential claims impacted, if a work around is available, etc.*

*Problem Priority Classification:*

*Is this problem prioritized as an emergency, 1, 2, 3, 4, or 5.*

*Release Options:*

*Explain the options for scheduling and implementing the fix.*

*Technical Recommendation for Release timing:*

*Explain the recommended timing for installing the release.*

**CMS System Maintenance Leads**

*Maintainers must forward schedules and documentation of all changes as required in the memorandum to your CMS maintenance lead as indicated below. If your current process is to forward this information to your project officer, continue to do so. Your CMS maintenance leads will advise you of backup staff.*

**40.3 - Shared System Testing Requirements for Maintainers, Beta Testers, and Contractors**

**(Rev. 6, 05-28-04)**

*Medicare requires implementation of a limited number of shared systems that must be used by all FIs and carriers. This eliminates the need for each Medicare Contractor to repeat development of the basic system.*

*CMS requires that the shared system quarterly release be subjected to the complete testing life cycle prior to production release. The goal is to ensure that all changes function as intended and that the implementation of changes does not degrade or otherwise unintentionally affect existing system capability and function prior to implementation. This requires that the shared system be subjected to all levels and types of testing including unit testing, integration testing, systems testing, functional testing, interface testing, performance testing, regression testing, and operational testing. Definitions are provided in subsection 40.3.9.*

*The Shared System Maintainer and the Medicare Contractor each have specific roles in testing the shared system quarterly release. Additionally, CMS contracts with an FI, Carrier, DMERC, and CWF Host to act as a Beta tester for the FISS, MCS, VMS-B/DME, and CWF systems respectively.*

*This section identifies the testing responsibilities for each organization to ensure that each shared system quarterly release satisfies all CMS requirements. All organizations shall have processes in place to meet these requirements. Testing activities will generally begin 3 to 4 months in advance of the release date, particularly for shared system maintainers and the CWF maintainer.*

### **40.3.1 – Maintainers and Beta Testers – Required Levels of Testing** **(Rev. 6, 05-28-04)**

*Review subsection 40.3.9, Definitions, for a description of key testing terminology.*

- 1. Maintainers of a Shared System or the CWF shall plan and execute all the essential levels of testing. At a minimum this includes Unit testing, Integration testing, System testing, and Regression testing. Maintainers are also responsible for performing Interface Testing.*
- 2. Beta testers may initiate testing at the integration level, but are primarily dedicated to testing at the system level, including regression testing. Beta testers are also responsible for performing Interface Testing, which includes full data exchanges between the Shared system, CWF, and other systems (e.g., HIGLAS when implemented).*
- 3. Maintainers and Beta testers shall maintain a test environment that enables system-testing activities to replicate the production environment, as closely as required to effectively test. CMS provided all Beta testers with a date simulation tool to facilitate executing test cases with future dates (e.g., service dates, admission dates) without turning off edits or altering effective dates in the test environment.*

### **40.3.2 – Minimum Testing Standards for Maintainers and Beta Testers** **(Rev. 6, 05-28-04)**

- 1. The Shared System Maintainer (SSM), the CWF Maintainer (CWFM), and the designated shared system Beta tester shall fully test the quarterly release to ensure it is ready to be elevated to production. For the quarterly release to be considered fully tested, all the requirements contained within the release must be tested. Maintainers and Beta testers must be able to demonstrate the degree to which each discrete requirement within a CR has been tested and by which test cases. It is therefore mandatory that the testers maintain traceability between test cases and the discrete requirements being implemented in the release. Additionally, for each CR or transmittal under test, the Maintainer and Beta tester must ensure that each discrete requirement specified in the **Business Requirements** section of any CR/transmittal has been fully tested. The Maintainer and Beta tester shall specifically:*
  - Maintain a repository of Test Requirements against which all test cases must be traced.*
  - Prepare and execute a set of Test Cases that demonstrate the requirements were correctly implemented for all change requests within the quarterly release.*

- *Maintain traceability between each Test Case and the requirement that the case was designed to test.*

2. *The Maintainer and Beta tester shall distinguish each Test Requirement with a unique Requirement Identifier. The Requirement Identifier must be a number or qualifier preceded by the CMS CR number and SSM CR number, separated by dashes. The format of the Requirement Identifier is: [SSM CR No.]-[CMS CR No.]-[Requirement No.], where:*

- *SSM CR No. – is a number that identifies a CMS mandate or user change request under test. Free form text can also be used to identify changes not associated with maintainer CR numbers, e.g., “Regression” to indicate regression testing. Avoid spaces and use underscore symbol “\_” instead. Dashes not allowed within this number; they are reserved for separation between the individual numbers in the Requirement Identifier.*
- *CMS CR No. – is a minimum 4-digit number that identifies the CMS CR associated with the maintainer CR number. If no CMS CR is associated with a maintainer CR, use “0000”. Dashes are not allowed. Avoid spaces and use underscore symbol “\_” instead. Dashes not allowed within this number; they are reserved for separation between the individual numbers in the Requirement Identifier.*
- *Requirement No. – is a number that uniquely identifies the requirement with the CR. For any requirement taken from the Business Requirements section of a CMS CR, use the actual number from the Requirement # column. Do not repeat the CMS CR number. Dashes not allowed within this number; they are reserved for separation between the individual numbers in the Requirement Identifier.*

*Example: Maintainer CR 22522 corresponds to CMS CR 2634. Business Requirement 2.8 was taken directly from CMS CR 2634. The Requirement Identifier would be 22522-2634-2.8*

3. *The Maintainer and the Beta tester shall complete Test Case specifications that include specific input situations and the expected results associated with a single test purpose. Each test case specification must include the following:*

- *A unique Test Case Identifier (which includes a cross-reference to the requirement in which the case is designed to test);*
- *The specific objective or purpose of the case;*
- *Input specifications (i.e., a description of the input situation[s]);*
- *Output specifications (i.e., a description of the expected results); and*
- *Intercase dependencies - in instances where the test results of one test case may impact other test cases, the test case specification must identify the other test case(s) and describe the relationship(s).*

*Refer to section 40.3.10, Test Case Specification Standard, for the specific format required to electronically maintain test cases.*

4. *All Test Cases must contain a unique Test Case Identifier. The CMS standard for the Test Case Identifier is the Requirement Identifier, followed by a number that uniquely qualifies the test case specification, separated by a dash.*

*The format of the Test Case Identifier is: [Requirement Identifier]-[Test Case Number], where:*

- *Requirement Identifier – is the actual identifier of the requirement being testing by the case.*
- *Test Case Number – is a number that uniquely qualifies the test case. This is generally a sequential number. This is necessary since more than one test case is often needed to test a single requirement. Dashes not allowed within this number; they are reserved for separation between the individual numbers in the Test Case Identifier.*

*Example: Two test cases were developed to test the implementation of Requirement 22522-2634-2.8 (see example above). The unique Test Case Identifier for the two test cases would be 22522-2634-2.8-01 and 22522-2634-2.8-02.*

5. *The Maintainer and the Beta tester shall document and execute **both** positive and negative test cases to ensure the requirements of the release are correctly implemented.*
  - *Positive test cases are required to ensure that the system is directly fulfilling the requirements as specified. One or more positive test cases are required for each requirement. As an example, if a program mandate effects a change for services beginning on July 1, a positive test case would include service dates in July or later and validate that the actual mandate was correctly implemented.*
  - *Negative test cases test cases are required to ensure that the system does not perform an incorrect action. As an example, if a program mandate effects a change for services beginning on July 1, a negative test case would ensure that implementing the mandate did not negatively impact claims with service dates prior to July 1. Unlike positive test cases, a negative test case may not be applicable to every requirement within a CR. Additionally, although due diligence might necessitate a negative test case, the need may be mitigated by an existing case in your regression test set.*
6. *The Maintainer and Beta tester shall document all test cases and the actual results for each test case electronically. Each test case and the associated results must be stored in a test management repository (i.e., TestDirector) and must at a minimum contain the data elements outlined in the CMS Test Case specification standard. See subsection 40.3.10 for the Test Case specification standard.*
7. *The Maintainer and Beta tester shall maintain a test log that provides a record of each test execution. Test Log requirements may be fulfilled by correctly using the TestDirector “run” feature as outlined in the Quarterly Release Test Management User Guide.*
8. *The Maintainer and Beta tester shall execute a full regression test set on their system for every quarterly release. Each testing entity shall perform regression testing within their designated testing window as outlined in subsection 40.3.7, Timeframe Requirements.*

9. *The Maintainer and Beta Tester shall perform interface testing.*

- *The Maintainer and Beta Tester shall validate that all output files are correctly created by the their system. The SSM and Beta Tester shall validate that their system can accept and correctly process all input files.*
- *The Shared System Maintainer and Beta Tester shall perform interface testing that includes full data exchanges (both ways) between the shared system and any principal claims processing adjudication or financial system. (e.g., the CWF and HIGLAS respectively). The Beta tester is required to perform data exchanges with HIGLAS after HIGLAS is implemented at Beta tester's data center.*
- *The Shared System Maintainer and Beta tester shall complete an integrated system test with the CWF. Each Maintainer and Beta tester shall coordinate the maintenance of test data baselines, such as beneficiary data, with the CWF Beta tester.*

### **40.3.3 – Testing Standards Applicable to all Beta Testers** (Rev. 6, 05-28-04)

1. *The FISS, MCS, VMS-DMERC, and VMS-B Beta testers shall complete integrated testing with the CWF Beta tester, using coordinated beneficiary data, in the execution of their test cases. All test cases involving CWF functionality (related to claims adjudication) must be executed in an integrated test with the CWF. This requires full data exchanges between testing entities including:*

- *Satellite files being sent from the shared systems to the CWF Beta tester; and*
- *Response files being sent from the CWF to the shared system Beta testers.*

2. *Each Beta tester shall:*

- *Utilize the standard CMS Test Management tool and repository to documents all test cases and results.*
- *Follow the procedures outlined in the Quarterly Release Test Management User Guide in order to complete the documentation of test runs and results.*

3. *The Beta tester shall review all Maintainer release documentation for completeness, accuracy, and usability. Any questions, problems, or issues with the documentation shall be forwarded to both the Maintainer and CMS.*

4. *The Beta Tester shall conduct performance testing to reasonably assure that the system provides acceptable response times, throughput rates, and processing windows and can accommodate production workloads.*

5. *The CMS testing requirements outlined in section 40.3 may require the Beta tester to test a specific type of bill, specialty, or claim situation for which they do not possess the required level of expertise. In these instances, the Beta tester must partner with a Medicare Contractor that possesses both the expertise and capabilities to test the specialty or claim type. As an example, should a Beta tester not have the operational capability or expertise to*

*process Home Health claims, they are expected to partner with an RHHI to complete the required HHA testing. Ultimately, the Beta tester is responsible for ensuring all test cases are exercised. Any partnerships that are established to complete the testing requirements, shall be arranged and managed by the Beta tester.*

#### ***40.3.4 – Testing Requirements Applicable to the CWF Beta Tester (Rev. 6, 05-28-04)***

*The CWF Beta tester shall act as a test host and exchange data with entities testing the FISS, MCS, VMS-DMERC, and VMS-B shared systems. The testing entities include all shared system maintainers and the shared system Beta testers.*

#### ***40.3.5 – Contractor (User) Testing Requirements (Rev. 6, 05-28-04)***

*Medicare Contractors are not mandated to prepare and execute test cases that cover Medicare business requirements implemented within the base system in shared system and CWF quarterly releases. Maintainers and Beta testers are fully responsible for testing the base functionality. The Medicare Contractors (users) shall first test their local/unique components and coding, and secondly conduct a limited, end-to-end, operational test.*

- 1. Contractors shall fully test their local components and unique coding prior to production implementation of the quarterly release. This testing is applicable for all local components and local coding modified since the previous quarterly release.*
  - A. Contractors shall test any system components they maintain and implement to support claims processing in addition to the base system. This includes front-end and back-end components such as those for EDI entry and translation, EDI outbound processing, and printing (e.g., MSN generation).*
  - B. Contractors shall test any unique source code they implement within the shared system.*
  - C. Contractors shall test changes they make to user control files, facilities, and tables in order to implement new Medicare policy or business rules.*
  - D. A Medicare Contractor shall fully test any shared system functionality that was:
    - Developed by the shared system maintainer solely for them, or*
    - Developed by the shared system maintainer under a special project in which they were the exclusive participant.**

*An example would be a Carrier working with CMS on a special demonstration project. In this example the carrier shall fully test the shared system functional that was implemented for the demonstration project.*

- 2. Contractors shall complete a limited end-to-end operational test that incorporates the shared system release, integrated with their other claims processing components. These components include the front-end for claims receipt, translators, the CWF, the financials, and back-end EDI and report generation. The test must ensure that processing is contiguous from claims entry, to claims adjudication, and ultimately remittance and Medicare Summary Notice*

*generation. Through contiguous processing, the interfaces between all key claims processing components must be exercised. The banking system interfaces such as National Clearing House (NCH) transfers need not be exercised. The test is limited in the number of test cases that are required, since maintainers and Beta testers are testing the base functionality of the shared system.*

*A. Contractors shall ensure that the integrated systems software can complete cycles without system abends and produce the expected output. The Medicare Contractor shall ensure their operational test:*

- Exercises all claims entry points not fully incorporated in the base system such as paper and EMC front-end components;*
- Includes all allowable standard electronic formats and versions;*
- Includes a variety of claims types; and*
- Includes all components that support their claims workload and interfaces to the shared system.*

*B. The operational test shall include the most recent shared system release received prior to the initiation of the test. The Medicare Contractor shall initiate the test as required to ensure its completion and the reporting of any problem prior to production implementation.*

*3. CMS strongly encourages the shared system user community to promote:*

- Standardizing their shared system nationally,*
- Centralizing any table maintenance that implements national Medicare policy at the system maintainer level, and*
- Minimizing local variations and coding.*

*4. Medicare Contractors may perform additional testing on the shared system or duplicate Beta testing tasks as time permits. At the discretion of their Regional Offices, Medicare Contractors may be required to separately document any testing they perform in addition to their mandated testing.*

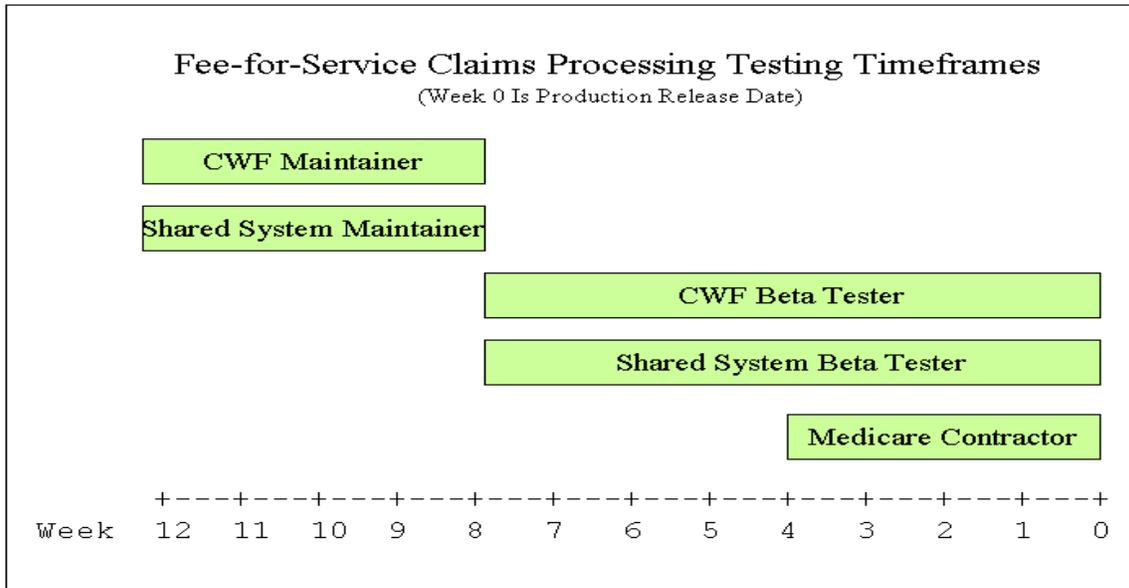
#### ***40.3.6 – Testing Requirements Applicable to all CWF Data Centers (Hosts) (Rev. 6, 05-28-04)***

*Each CWF data center or CWF sector host shall:*

- Forward all satellite software and documentation to the satellites (contractors) to which they serve as the primary production host;*
- Install the CWF quarterly release software in a designated test region;*
- Make the designated test region available their satellites (users) for testing;*
- Coordinate test data, such as beneficiary data, with the user testers;*
- Process all satellite files submitted by the users and return all corresponding reply files generated for users;*
- Report release problems to the CWF Maintainer and CMS; and*
- Verify with CMS that each of its satellites submitted at least one test file during user testing.*

**40.3.7 – Timeframe Requirements for all Testing Entities**  
**(Rev. 6, 05-28-04)**

The SSM, CWF, Beta tester, and Medicare Contractor shall operate under the testing timeframes shown below for each quarterly release:



1. The Medicare **Contractor** or **User** testing period shall begin four weeks prior to production implementation.
2. The **Beta** testing period shall begin eight weeks prior to production implementation. The CWF and shared system Beta testers shall have an exclusive four-week testing timeframe prior to the initiation of user testing.
  - The Beta tester shall complete a functional System Test and Regression Test before the shared system is released to the User community.
  - Beta testing must also continue through the User testing period. The Beta tester may initiate performance testing during the user testing period.
3. Exclusive CWF and SSM testing shall continue until Beta testing is initiated eight weeks prior to production implementation. The SSM and CWF shall complete a Unit Test (on all components), Integration Test, System Test, and Regression Test prior to distributing the shared system release to the designated Beta tester. For all integration, system, and regression testing, the SSM shall use the most recent version of any third party or CMS provided software components (e.g., Pricer, OCE, MCE, Grouper) they are provided. The SSM shall continue testing beyond the exclusive maintainer-testing window due to the late receipt of some third party or CMS provided software components such as the Pricers and OCE.

**40.3.8 – Testing Documentation Requirements**  
**(Rev. 6, 05-28-04)**

1. *The SSM, Beta tester, and Medicare Contractor shall maintain documentation that fully demonstrates the requirements of this transmittal were met for each quarterly release. At a minimum the SSM, Beta tester, and Medicare Contractor shall maintain the following test documentation to demonstrate full compliance:*
  - *Test Requirement and Test Case repository with traceability;*
  - *Test Log;*
  - *Test Status for the execution (run) of each test case (i.e., Pass, Fail, Not Run);*
  - *Actual Results for the run of each test case in which the actual results did not match the expected results; and*
  - *Documented proof of each test run, i.e., screen shots, scheduler job logs, etc.*
2. *The SSM, Beta tester, and Medicare Contractor shall:*
  - *Maintain all test documentation for the four quarterly releases prior to the current release under test. The documentation must be available for review by CMS (or its agent).*
  - *Document all software defects (problems) within the CMS specified repository such as INFOMAN.*
3. *The SSM shall communicate all confirmed software defects (problems) and fixes directly to CMS in writing through their CMS Maintenance Lead or other designee as specified by the CMS Project Officer.*
4. *The Medicare Contractor shall provide any testing documentation to their CMS regional office upon request.*

*Additional requirements for selected shared system and CWF maintainers, Beta test sites, and CWF hosts may be contained in these organizations' individual contracts. Electronic screen shots may be incorporated/attached into the results of TestDirector has proof of online results. Test Log requirements may be fulfilled by correctly using the TestDirector “run” feature as outlined in the Quarterly Release Test Management User Guide.*

### **40.3.9 – Definitions**

**(Rev. 6, 05-28-04)**

*These definitions are provided to ensure common understanding.*

*Base System - The FISS, MCS, VMS, or CWF system, which is routinely released by the Shared System Maintainer (i.e., Ark-FISS, EDS, ViPS) to their respective user community prior to any user customization. This includes all components released by the Maintainer, including but not limited to the claim adjudication subsystem, the financial subsystems, and other integrated components (i.e, Pricer, OCE, MCE, Grouper).*

*Functional Testing – Testing to ensure that the functional requirements have been met.*

*Integration Testing – Testing combinations of interacting software components the make up parts of a system.*

Interface Testing – Testing conducted to evaluate whether subsystems or systems pass data and control to one another correctly.

Local Components – A Local Component as referenced in section 40.3 is any component or module that supports Medicare claims processing, but is not part of the Base System.

Maintainer – The Maintainer is an entity to which CMS directly contracts to maintain a Medicare claim processing shared system (FISS, MCS, or VMS-DME) or the Common Working File (CWF) system. The Maintainer, as referenced in section 40.3, does **not** refer to an entity to which a Medicare Contractor (Carrier, Fiscal Intermediary, or DME Regional Contractor) subcontracts to operate their data center or perform other claim processing support activities.

Operational Testing – Testing conducted to evaluate a system in its operational environment. Testing to ensure that the aggregate operational systems and their interfaces can be operated securely with the instructions provided.

Performance Testing – Testing that applies heavy transaction and processing loads to the system to ensure that response times, throughput rates, and processing windows remain acceptable and can accommodate production workloads.

Regression Testing – Testing conducted on a system or components to verify that modifications have not caused unintended effects and that the system or components still complies with its requirements.

Regression Test Set – A set of selectable test cases designed to exercise a system over its functional capabilities and assure that it still works properly after changes have been applied.

Requirement Identifier – A unique number assigned to each requirement comprised of the Shared System Maintainer CR Number, the CMS CR Number, and an alphanumeric element to uniquely qualify each requirement. For testing purposes CMS requires that each Test Case Identifier incorporate the Requirement Identifier to which it is traced.

Stress Testing – Testing that applies a steadily increasing load to the system until it reaches the point where performance degrades to unacceptable levels.

System Testing – Testing to discover any incorrect implementation of the requirements or incompatibilities in the software/hardware environment. System testing includes functional testing, performance testing, and operational testing.

Test Case Specification – A description of an input situation and of the required results associated with a specific test objective or purpose.

Test Case Identifier – A unique identifier assigned to each test case.

Test Log – A chronological record of relevant detail about the execution of tests. Relevant details include run date, run time, test status, and actual results.

Test Requirement – A specific requirement that is under test and to which one or more test cases are traced. Test requirements may be derived from various types of requirements i.e., business

functional requirements, performance requirements etc. Note: Any well-written requirement that is “testable” may be considered a Test Requirement. Any requirement contained in the Business Requirements section of a CR or transmittal, also constitutes a test requirement.

Test Set – A collection of test cases that have a common usage.

Unit Testing – The testing of individual units (i.e., software components, modules) or groups of related units. It is the lowest level of testing and is usually performed by programmers. Unit testing may be both functional (requirements oriented) and structural (i.e. logic oriented, code coverage oriented).

### **40.3.10 - Test Case Specification Standard (Rev. 6, 05-28-04)**

**Purpose:** This standard establishes a controlled outline for the contents and presentation of a Test Case Specification used by the standard system maintainers and the Beta testing contractors.

**Applicability:** This standard is applicable to all Test Case Specifications developed by the standard system maintainers and the Beta testing contractors.

<b>Data Element</b>	<b>Description</b>	<b>Allowable Values or Format</b>	<b>Comments</b>
<b>Test Case Specification Identifier</b>	Multi-part indicator that uniquely identifies the test case specification.	See Test Case Specification Identifier Standard.	
<b>Test Purpose</b>	A free form field that captures the intent of the test and identifies any key components of the test, e.g., specific codes.	See attached example.	
<b>Input Specification</b>	A free form field that captures critical information used to exercise the system functionality. Information could be grouped into the following topics: <ul style="list-style-type: none"> <li>• Claim Data Requirements</li> <li>• Claims History</li> <li>• Beneficiary Information</li> <li>• Provider Information</li> </ul>	See attached example.	
<b>Intercase Dependencies (Predecessor Transaction Identifier)</b>	The test case specification identifier of the transaction that must be entered into and processed by the system prior to processing the transaction described by the test case specification.	See Test Case Specification Identifier Standard.	

<b>Output Specification</b>	<i>A free form declarative statement that identifies the expected results from performing all the steps, as a collection, within the test.</i>		
<b>Test Type</b>	<i>A one-character indicator to identify whether the test is positive or negative.</i>	<i>P = Positive Test N = Negative Test</i>	<b>TestDirector Plan Tab</b>  <i>(Required User Defined Fields)</i>
<b>Originator</b>	<i>A one-character indicator to identify the originating entity (designer) of the test case.</i>	<i>B = Beta C = CMS/QRTM M = Maintainer</i>	
<b>Test Status</b>	<i>Summary indicator for a test case.</i>	<i>PS = Passed FA = Failed NR = Not Run IN = Incomplete ID = Invalid Data IC = Invalid Case</i>	<b>Required Test Execution (Run) Elements</b>
<b>Test Results</b>	<i>Free form declarative statement of actual results for a test case when the actual results do not match the expected results.</i>		

*Optional Information: Industry best practices demonstrate that additional granularity may be necessary to document discrete key test actions that should be executed and documented. These items are referred to as test steps. A test case specification may have one or more test steps. When documenting test steps, the following standard applies:*

<b>Step Number</b>	<i>Unique identifier for each test step.</i>	<i>“Step n” Where “n” is a sequential counter for each step starting at 1.  There is at least one test step in each test case specification, but usually contains multiple test steps.</i>	<b>Optional Test Case Elements</b>
<b>Step Description</b>	<i>A free form declarative statement that identifies the action taken to perform the test. The step description statement usually begins with a verb.</i>		
<b>Expected Step Results</b>	<i>A free form declarative statement that identifies the expected results from performing the associated step description.</i>		

**Example #1**

<b>Test Case Identifier</b>		4393-2342-12.1-001
<b>Test Purpose</b>		To confirm that the FI claims processing systems accept, process and assign a reason code to Hospital claims with services denied based on a Local Medical Review Policy (LMRP) submitted on Type of Bill (TOB) 141 (Hospital other or referred diagnostic services; admit through discharge), generating Medicare Summary Notice (MSN) message 15.20 (The following policies were used when we made this decision) auto-filling LMRP identification (ID) number L481 (Breast Imaging; Mammography/Breast Echography [Sonography]/Breast) associated with the edit for each fully denied service.
<b>Input Specification</b>	<b>Claims History</b>	Mammography service previously rendered and paid $\leq$ 11 months
	<b>Beneficiary Information</b>	Beneficiary has elected English as primary language.  Female  Age: $\geq$ 40
	<b>Provider Information</b>	Provider Number Range: XX0001 - XX0999
	<b>Claim Data Requirements</b>	TOB: 141 (Hospital other or referred diagnostic services; admit through discharge) Revenue Code #1: 403 (Other imaging services; screening mammography) Units #1: 1 HCPCS Code #1: 76091 (Mammography; bilateral) Revenue Code #2: 403 (Other imaging services; screening mammography) Units #2: 1 HCPCS Code #2: 76092 (Screening mammography; bilateral [two view film study of each breast]) Diagnosis Code: V76.12 (Other screening mammography)
<b>Intercase Dependencies</b>		None
<b>Output Specification</b>		Claim will be assigned a reason code indicating services denied based on LMRP ID# L481, generating MSN message 15.20.
<b>Test Type</b>		P
<b>Originator</b>		C
<b>Test Status</b>		PS
<b>Test Results</b>		Claim was assigned appropriate reason code

**Example #2**

<b>Test Case Identifier</b>	<i>4419-2825-5.2-001</i>	
<b>Test Purpose</b>	<i>To confirm that the FI claims processing systems accept, process, and assign reason code 30 (Payment adjusted because the patient has not met the required eligibility, spend down, waiting or residency requirements) to Inpatient Hospital claims submitted on Type of bill (TOB) 111 (Hospital Inpatient Part A; admit through discharge) with Dates of Service (DOS) on 01/01/2004 when a beneficiary is not lawfully present in the United States.</i>	
<b>Input Specification</b>	<b>Claims History</b>	<i>None</i>
	<b>Beneficiary Information</b>	<i>Beneficiary must be unlawfully present in United States. Beneficiary elected English as primary language</i>
	<b>Provider Information</b>	<i>Provider Number Range = XX0001-XX0999</i>
	<b>Claim Data Requirements</b>	<i>TOB = 111 DOS = 01/01/2004</i>
<b>Intercase Dependencies</b>	<i>None</i>	
<b>Output Specification</b>	<i>Claim will be assigned reason code 30 indicating beneficiary is not lawfully present in the United States, generating MSN message 5.7 (Medicare payment may not be made for the item or service because on the date of service, you were not lawfully present in the United States).</i>	
<b>Test Type</b>	<i>P</i>	
<b>Originator</b>	<i>C</i>	
<b>Test Status</b>	<i>PS</i>	
<b>Test Results</b>	<i>Claim was assigned appropriate reason code</i>	