Summary Care Record

Clinical Message Validation Process Guide

Amendment History:

|  |  |  |
| --- | --- | --- |
| Version | Date | Amendment History |
| 0.7.3 |  | First draft to be distributed for comments |
| 0.8 |  | Final draft for approval |
| 1.0 | 30/10/2009 | Final version incorporating comments from Mike Frederick and Leo Fogarty. |

Forecast Changes:

|  |  |
| --- | --- |
| Anticipated Change | When |
| 1.1 – Update version reflecting changes to MQC usage based on experience. | 18 December 2009 |

Reviewers:

This document must be reviewed by the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title / Responsibility | Date | Version |
| Leo Fogarty |  | Clinical Safety Officer, SCR Programme | 20/10/2009 | 0.8 |
| Angela North |  | SCR Release & Assurance Manager |  |  |
| Mike Frederick |  | SCR Release Manager (to-be CMV Manager) | 27/10/2009 | 0.8 |

Approvals:

This document must be approved by the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title / Responsibility | Date | Version |
| Leo Fogarty |  | Clinical Safety Officer, SCR Programme |  |  |
| Angela North |  | SCR Release & Assurance Manager |  |  |

Distribution:

This document is intended to be read by the following people:

* Members of the CMV team
* Release Managers responsible for projects that may require CMV testing
* Optionally by Clinical System Suppliers involved in CMV testing

Document Status:

This is a controlled document.

Whilst this document may be printed, the electronic version maintained in FileCM is the controlled copy. Any printed copies of the document are not controlled.

Related Documents:

These documents will provide additional information.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref no | Doc Reference Number | Title | Version |
| 1 | NPFIT-SHR-QMS-PRP-0015 | Glossary of Terms Consolidated.doc | <enter latest> |

Glossary of Terms:

List any new terms created in this document. Mail the NPO Quality Manager to have these included in the master glossary above [1].

|  |  |  |
| --- | --- | --- |
| Term | Acronym | Definition |
| Personal Spine Information Service | PSIS | Central database storing data related to the Summary Care Record |
| Clinical Message Validation | CMV | An assurance testing activity conducted by the Summary Care Record programme to confirm that all PSIS messages are clinically safe. |
| Summary Care Record | SCR | The component of a patient’s electronic care record providing a summarized history of all significant clinical information and encounters. |
| Summary Care Record Application | SCRA | A web-based application providing view-only access to a patient’s Summary Care Record. |
| Health Level 7 version 3 | HL7v3 | A standard XML-based message format maintained by the HL7 Foundation, used for the reliable representation of clinical information. |
| Electronic Business by Extensible Mark-up Language | ebXML | A messaging specification designed to provide reliable exchange of messages between heterogeneous systems. It is typically used in the CFH programme for critical messages such as updates. |
| Multi-Purpose Mail Extensions | MIME | An internet standard used for structuring email messages, but also used to structure content sent using HTTP such as web services. |
| Systematised Nomenclature of Medicine | SNOMED | A library of medical terms designed for systematic use, covering most diagnoses, procedures, observations. |
| Dictionary of Medicines and Devices | dm+d | A dictionary containing unique identifiers and associated textual descriptions for medicines and medical devices. |
| Mercury Quality Centre | MQC | A web-based testing tool, designed to manage product test cycles. Provides Requirements Traceability, Test Planning, guided Test Execution and Defect Management. |

Contents

[1 About this Document 7](#_Toc244679985)

[1.1 Purpose 7](#_Toc244679986)

[1.2 Audience 7](#_Toc244679987)

[2 Introduction 8](#_Toc244679988)

[2.1 The NHS Summary Care Record (NHS SCR) 8](#_Toc244679989)

[2.2 SCR Compliance Modules 8](#_Toc244679990)

[2.3 Objective of Clinical Message Validation 9](#_Toc244679991)

[2.4 Overview of Clinical Message Validation 9](#_Toc244679992)

[2.5 Need for Clinical Message Validation 10](#_Toc244679993)

[2.6 CMV Approach 11](#_Toc244679994)

[3 CMV Process – Step by Step Guide 12](#_Toc244679995)

[3.1 Overview 12](#_Toc244679996)

[3.2 Step 1 – Initial Engagement 14](#_Toc244679997)

[3.3 Step 2 – Design Engagement 16](#_Toc244679998)

[3.4 Step 3 – CFH/Supplier Engagement 18](#_Toc244679999)

[3.5 Step 4 – Clinical System Familiarisation 20](#_Toc244680000)

[3.6 Step 5 – Stand Alone Testing 22](#_Toc244680001)

[3.7 Step 6 – Integrated Testing 25](#_Toc244680002)

[3.8 Step 7 – Closure 28](#_Toc244680003)

[4 Appendix A – Mercury Quality Centre 30](#_Toc244680004)

[4.1 Introduction 30](#_Toc244680005)

[4.2 Overview 30](#_Toc244680006)

[4.3 Requirements 31](#_Toc244680007)

[4.4 Creating Tests 33](#_Toc244680008)

[4.5 Test Lab 35](#_Toc244680009)

[4.6 Defect Management 37](#_Toc244680010)

[4.7 Worked Example – CMV Testing of the Initial GP Summary Upload 44](#_Toc244680011)

[5 Appendix B – Planning Guidelines 60](#_Toc244680012)

[5.1 Objectives 60](#_Toc244680013)

[5.2 Planning Assumptions 60](#_Toc244680014)

[5.3 High-Level Demand Management Process 60](#_Toc244680015)

[5.4 Estimating Model 61](#_Toc244680016)

# About this Document

## Purpose

This document is intended to serve as a guide to the Clinical Message Validation (CMV) process, conducted as part of the Summary Care Record programme assurance activity.

The Summary Care Record programme is concerned with the exchange of clinical messages with the Personal Spine Information Service (PSIS). Given the degree of transformation that such messages undergo in being sent between systems it is essential to assure that clinical systems are ‘clinically safe’ in this context and that message semantics are maintained. This is the purpose of CMV.

The following sections will provide a guide to all aspects of the CMV process, including background, requirements, test plans, dependencies, inputs, the process itself, and required outputs / sign-offs.

This document serves as a guide to the CMV process only. Other supporting documents and tools will be produced to support the process and provide detail.

## Audience

The proposed audience for this document is as follows:

* Summary Care Record programme resources involved in CMV testing who need to apply a consistent process to all CMV activities
* CFH staff who require an understand of CMV for test planning purposes
* CFH clinical and technical assurance staff who wish to understand the coverage of, or overlap with CMV testing
* Clinical System Suppliers who will be subject to CMV and wish to understand the scope of this work and their contribution to it.

# Introduction

## The NHS Summary Care Record (NHS SCR)

The NHS Summary Care Record (SCR) is one element of the NHS Care Records Service (NHS CRS), the other element being the Detailed Record. The availability of the SCR can provide benefits to both staff and patients by giving healthcare staff faster, easier access to reliable information so they can provide more effective treatment to patients.

A patient's SCR can be composed of the summaries of clinical encounters in many healthcare settings, for example GP Summaries, or Out of Hours encounters. Initial efforts are focussed on the GP Summary, as GP systems typically have the most complete and current record of a patient’s general health. Initially, an SCR will contain only key health information such as details of allergies, current prescriptions and adverse reactions to medicines, provided as part of an ‘initial upload’ of a GP Summary. For every subsequent occasion that the patient uses any NHS service a summary of that encounter may be added to their Summary Care Record.

Once uploaded to PSIS, SCR records will be accessed either by clinical systems or via the Summary Care Record Application (SCRa), a web-based application providing view-only access to Clinical data.

## SCR Compliance Modules

### SCR Compliance Module v1 (SCR1)

The SCR1 compliance module is scoped to only include GP Summaries. GP records are regarded as being the foundation of a patient’s record, to which other clinical settings will add additional information as described in section 2.2.2.

### SCR Compliance Module v2 (SCR2)

SCR2 includes the same GP Summary component as SCR1 but covers the following additional clinical settings and functions:

* Admissions
* Ambulance
* Diagnostic Imaging
* Inpatient Discharge
* Emergency Department
* Health and Social Care Integration
* HealthSpace
* Mental Health
* NHS Direct
* Out of Hours
* Outpatients
* Pathology
* Sealing

## Objective of Clinical Message Validation

The purpose of SCR is to share clinical information between heterogeneous software solutions. This poses an important question concerning the reliability and safety of the resulting transformations and transportations of the SCR content. The objective therefore of CMV assurance activity is to assure that the contents of SCR messages are:

* Clinically compliant with the requirements
* Clinically safe

## Overview of Clinical Message Validation

### SCR Message Structure

CMV is concerned with assuring the safety and accuracy of SCR messages exchanged between clinical systems and PSIS. All messages exchanged with PSIS use the HL7v3 messaging standard, with either a web service (https) or ebXML transport wrapper as shown in Figure 1.

Figure - Construction of HL7v3 messaging using ebXML and/or web services

MIME Part

MIME Parts

hl7:MessageWrapper

SOAP-ENV:Envelope

SOAP-ENV:Body

SOAP-ENV:Header

eb:MessageHeader

eb:Manifest

hl7:ControlAct

eb:Acknowledgement

eb:ErrorList

eb:other

hl7:Payload

Communications Protocol (HTTPS)

The actual clinical content of the message is contained in the inner HL7 payload element, carried as part of the second MIME part of the HL7 message. Nothing outside this payload contains clinical data.

CMV is concerned only with validating clinical content, meaning and interoperability of the inner payload, as described below. The assurance of all other technical parts of the message is conducted by other CFH teams such as Technical Assurance Group.

## Need for Clinical Message Validation

Clinical data undergoes a number of automated transformations in the process of being converted to an HL7 message, uploaded to PSIS and displayed (e.g. via SCRa):

1. Patient Records are stored in the local clinical system with both human-readable text and clinically coded statements.
2. Human-readable text is extracted and converted to HTML using ‘Source-to-Target’ mapping tables.
3. Clinically coded statements are extracted and converted to SNOMED using mapping tables (e.g. READ<>SNOMED tables).
4. Drugs and Medications may also be mapped using the ‘dm+d’ dictionary.
5. The converted, standardised HL7 message is sent to PSIS and rendered on SCRa and other systems.

In order to ensure the preservation of clinical meaning within SCR, the following must be taken into account:

* The SCR extract process will take clinical information from a clinical system which will have its own idiosyncratic features and internal structures for representing clinical concepts. This is translated into the format required for the SCR message and stored on PSIS and rendered for display in SCRa and other systems. It is essential to ensure that the clinical meaning of the information is preserved throughout the transformation process.
* The SCR message contains both coded clinical concepts, and a ‘Presentation Text’ representation of the whole summary. Assurance is required that the text is an equivalent representation of the coded sections of the summary i.e. that the clinical system code (e.g. READ) accurately maps to an equivalent SNOMED code in the PSIS message, and also that Medications and Devices are appropriately mapped as per the dm+d dictionary.
* Assurance is also required that all system constructs and variations of clinical data possible within the clinical system are covered. This will require a complete set of test cases to be built which cover all variations within the system.

## CMV Approach

The generic approach to Clinical Message Validation can be summarised as follows:

1. Create test patient records in the subject clinical system
2. Generate SCR messages using the clinical system and capture the XML messages
3. Compare the clinical textual content of the XML message, local clinical system, and Summary Care Record Application (SCRa) to ensure semantics are maintained.
4. Compare the clinically coded content of the XML message to the source clinical system to ensure clinical codes are correctly mapped from source codes (e.g. READ) to SNOMED.

An informal check also carried out is to validate the syntactic envelope i.e. does the XML conform to the HL7 definitions. No defects are raised on this latter scrutiny process because it is not a formal part of CMV testing

There is a significant amount of preparation involved in performing CMV testing, including demand planning, system familiarisation, and test script preparation. Following completion of CMV testing, a requirements and defect management activity is also required.

# CMV Process – Step by Step Guide

## Overview

To ensure that future CMV workload can be effectively planned and managed, as well as assuring consistent results, the CMV process has been defined as a seven step model:

|  |  |
| --- | --- |
| *Initial Engagement* | Covers initial conversations between CFH release managers and the CMV team to confirm demand, scope, and expectations. |
| *Design Engagement* | Review of initial designs received by the release manager from suppliers, to identify any points of interest that may impact the CMV approach and scope |
| *CFH/Supplier Engagement* | A short activity to introduce or review CMV with the supplier team, and discuss process issues such as defect management. |
| *Clinical System Familiarisation* | A short period of testing allowing the CMV resource to become familiar with the supplier’s system prior to executing formal testing. |
| *Stand-Alone CMV Testing* | Formal CMV testing of the supplier product using a stand-alone version of the system to generate HL7 XML content for validation. Spine responses are not tested. |
| *Integrated CMV Testing* | The final, formal stage of testing including the exchange of messages with a test (sandpit) spine instance |
| *Project Closure* | Activities required to formally close the CMV testing and ensure that all defects are addressed, and to confirm sign-off from key resources. |

The following sections explain each step, to be executed in sequence, with the following key information:

* Process Diagram
* Objective
* Prerequisites
* Process
* Outputs
* Roles and Responsibilities (RACI Matrix)
* Estimated Duration
* Tools Required

Roles and Responsibilities are expressed using RACI matrices, where responsibilities are defined as follows:

**R**esponsible: Those who do the work involved in achieving the task

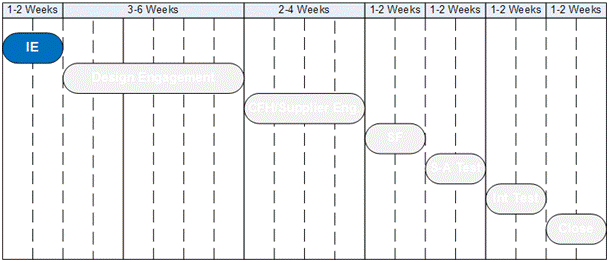
**A**ccountable: Those who are ultimately responsible for correctness of the completed task

**C**onsulted: Those whose opinions are sought on the task

**I**nformed: Those who are informed about progress or end results of a task

## Step 1 – Initial Engagement

### Overview Diagram



### Objective

To ensure clinical safety is maintained the CMV team must plan for adequate resource and time for each software release in scope. Insufficient time or resource may lead to a lower quality of testing and increased risk. The objectives of Initial Engagement are to:

* Provide structure for introducing new releases, to ensure CMV is correctly resourced and executed.
* Ensure NHS CFH Release Managers (RMs) are knowledgeable about CMV.
* Gather high-level parameters to enable CMV estimating and planning

### Prerequisites

Initial Engagement requires the following pre-requisites to be met:

1. CMV Capacity Plan is up-to-date with all existing demand.
2. RMs have high-level information regarding scope and timeline for release.

### Process

Initial Engagement will be conducted as follows:

1. RMs contact CMV manager to discuss new demand. A ‘New Demand Meeting’ is scheduled using a standard agenda (see section 3.2.8).
2. At the meeting the CMV manager ensures that all parties understand the scope of and need for CMV, using the CMV Process Guide.
3. During the meeting, an approximate scope of required CMV testing is agreed for estimating purposes using an Information Request Form (see section 3.2.8), including delivery dates.
4. CMV manager updates the CMV Capacity Plan following the meeting, and raises early warning of any resource conflicts, risks and issues.
5. CMV provide RMs with a CMV Readiness checklist.

### Outputs

The Initial Engagement step will result in the following outputs:

* Updated CMV Capacity Plan
* Tailored CMV Readiness Checklist
* Update CMV Risks and Issues Log

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Arrange and conduct Initial Engagement Meeting | I | A |  | R |  |  |  |
| Update capacity plan, resolve conflicts |  | RA | I | C |  |  |  |
| Prepare tailored CMV Readiness Checklist | C | RA |  | C |  |  | I |

### Estimated Effort

Approximately 2 weeks elapsed:

* 1 week to arrange and hold New Demand Meeting
* 1 week to update CMV Capacity Plan

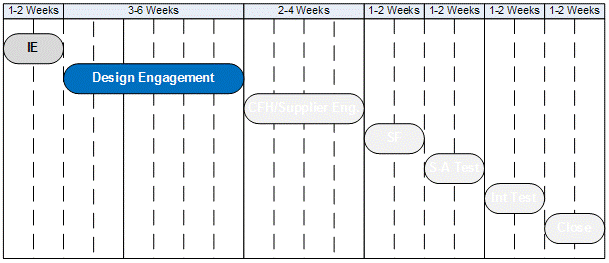
### Tools

Initial Engagement will require the following tools:

* CMV Process Guide
* Standardised New Demand Meeting agenda
* Information Request Form
* CMV Capacity Plan
* CMV Readiness Checklist

## Step 2 – Design Engagement

### Overview Diagram



### Objective

During the Common Assurance Process (CAP) Design and System Test Phase the Clinical System Supplier will deliver design documentation. The CMV team will review design deliverables to ensure that the functionality and any idiosyncrasies of the system are understood as early as possible. This allows the CMV team to:

* Revalidate high-level estimates generated in Step 1 – Initial Engagement
* Highlight any features that may require additional Test Scenarios to be developed (see Appendix A)
* Escalate early any clinical risks with the design.
* Become familiar with the system functionality and behaviour early in the lifecycle, and any deviations from previous builds.

### Prerequisites

Design Engagement requires the following pre-requisites to be met:

1. The Clinical System is included in the CMV Capacity Plan
2. Clinical Supplier has Authority to Proceed into Design for this system.
3. Clinical System Supplier has submitted all design documentation to NHS CFH for this system.

### Process

Design Engagement will be conducted as follows:

1. SCR or GPSoC Release Manager distributes Clinical System Supplier design documents to the CMV Lead.
2. CMV Lead distributes design documents to CMV team clinicians and the SCR Clinical Safety Officer.
3. CMV clinicians review documentation and provide formal comments to the SCR or GPSoC Release Manager
4. CMV clinicians develop a detailed Design and Risk statement indicating key system concepts, risks, issues and impact on Test Scenarios.

### Outputs

Design Engagement will result in the following outputs:

* Design and Risk Statement

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Submit Design documents as part of CAP Design and Test process, respond to comments |  | I |  | C |  |  | AR |
| Provide Design Documents to CMV Lead, collate comments |  | C |  | AR |  |  |  |
| Distribute Design documents for CMV review, provide comments | C | AR | C | I |  |  |  |
| Design and Risk Statement | A | R | C | I | I |  | I |

### Estimated Effort

From the submission of Design documents by the Clinical System Supplier, approximately 3-6 weeks elapsed. Further effort may be required depending on the outcome of reviews, and any additional document submission cycles.

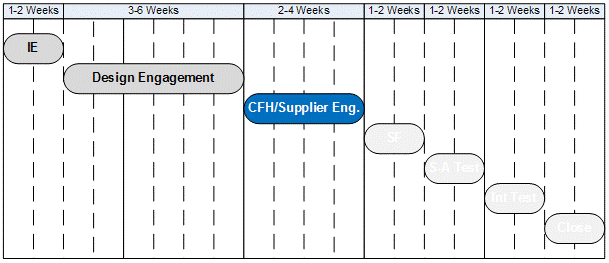
### Tools

Design Engagement will require the following tools:

* CMV Design and Risk Statement Template
* NHS CFH Standard Document Review Template
* CMV Test Scenarios Pack (using MQC)
* CMV Risk and Issues Log

## Step 3 – CFH/Supplier Engagement

### Overview Diagram



### Objective

Many supplier resources are unfamiliar with CMV testing, or it may have been some months or years since any previous CMV testing was performed. CFH/Supplier Engagement is intended to ensure that the supplier is knowledgeable about, and ready to support CMV. Subjects for discussion will include (but not be limited to):

* Environment requirements/availability
* Testing process
* Defect management

### Prerequisites

CFH/Supplier Engagement requires that the following pre-requisites to be met:

1. The Clinical System is included in the CMV Capacity Plan
2. Clinical System Supplier has Authority to Proceed into Design for this system.

### Process

Clinical System Familiarisation will be conducted as follows:

1. The NHS CFH CMV lead will, with the aid of the CFH Release Manager, schedule a suitable meeting with the Clinical System Supplier.
2. The NHS CFH CMV lease will send an agenda (see 3.4.8) to the supplier contact and ensure that the purpose of the Engagement session is clear.
3. The meeting will be conducted as arranged, using the agenda.
4. The NHS CFH CMV lead will follow up on any actions from the meeting.

### Outputs

CFH/Supplier Engagement will result in the following outputs:

* Updated defect management process if necessary
* Updated CMV Capacity Plan if necessary

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Schedule CFH/Supplier Meeting |  | C |  | AR |  |  | C |
| Distribute meeting materials |  | AR |  | I |  |  | I |
| Conduct CFH Supplier Meeting |  | AR |  | I |  |  | I |

### Estimated Effort

Elapsed time to organise and conduct the CFH/Supplier Engagement task will be about 2-4 weeks. Actual effort will be about 1-2 man days.

Further effort may be required depending on the outcome of reviews, and any additional document submission cycles.

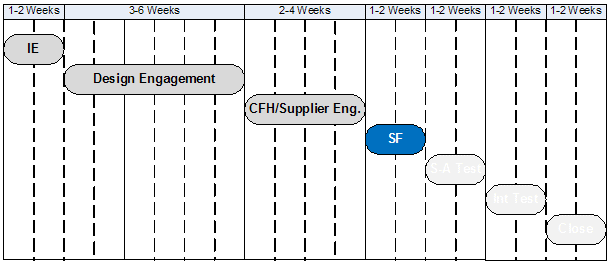
### Tools

CFH/Supplier Engagement will require the following tools:

* CFH/Supplier Engagement Meeting materials (agenda, minutes template etc)

## Step 4 – Clinical System Familiarisation

### Overview Diagram



### Objective

Understanding the Clinical Systems actual behaviour is key to ensuring that the CMV Test Plan is appropriate. Once a prototype Clinical System is available the CMV team will require access to the system in order to become familiar with its behaviour and features. Familiarity with the system will ensure that:

* Test execution is as efficient as possible
* Defects identified will be genuine and not the result of user error
* Test scripts are tailored where necessary based on actual system behaviour.

### Prerequisites

Clinical System Familiarisation requires the following pre-requisites to be met:

1. A representative version of the Clinical System is available in an accessible location.
2. CFH/Supplier Engagement has been conducted

### Process

Clinical System Familiarisation will be conducted as follows:

1. The Clinical System Supplier and the NHS CFH Release Manager will confirm that a working, representative version of the Clinical System is available
2. The NHS CFH CMV lead will contact an appropriate Clinical System supplier resource to arrange access to the Clinical System
3. The NHS CFH CMV lead will conduct free-roaming testing of the system using a checklist of functionality to explore.
4. The NHS CFH CMV lead will update any test scripts impacted by nuances of the Clinical System behaviour.

### Outputs

Clinical System Familiarisation will result in the following outputs:

* Updated Test Scripts/Scenarios (where necessary)

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical Systems Supplier |
| Confirm that a representative Clinical System is available for use |  | I |  | A |  |  | R |
| Arrange access to the Clinical System via an appointed Clinical System supplier contact |  | AR |  | I |  |  | C |
| Conduct Clinical System Familiarisation | I | AR |  | I |  |  | I |
| Update impacted Test Scripts | I | AR |  |  |  |  |  |

### Estimated Effort

Completing the Clinical System Familiarisation task will depend upon the level of change from any previous versions, and the knowledge of the CFH CMV lead. However, a reasonable estimate is 1 week of actual effort.

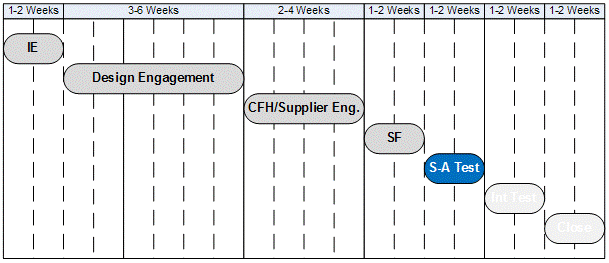
### Tools

Clinical System Familiarisation will require the following tools:

* CMV Clinical System Familiarisation Checklist/Process

## Step 5 – Stand Alone Testing

### Overview Diagram



### Objective

By the beginning of this activity the supplier will have been engaged and informed about CMV, the designs reviews, all test scripts updated, and the Clinical System itself will have been examined. Formal CMV testing will now commence.

The purpose of Stand-Alone testing is to conduct early CMV (during the Design and System Test phase ideally) to identify CMV defects before progressing the build to Integrated Testing.

### Prerequisites

Stand-Alone testing requires the following pre-requisites to be met:

1. Access to a working, representative build of the clinical system is available, with the ability to send spine messages using a test harness or other stubbed testing arrangement.
2. All test scripts are ready
3. Access to MQC (see section 3.6.8) is available from the testing location.

### Process

Stand-Alone testing will be conducted as follows:

1. The NHS CFH Release Manager will work with the clinical system supplier to ensure that a representative, stable build is available.
2. The CFH CMV Lead will arrange access to the test system with the Clinical System Supplier.
3. The CFH CMV Lead will run a series of tests using the Test Scripts documented in MQC. Defects shall be logged using MQC defect management
4. Tools such as an XML viewer and a SNOMED browser shall be used to examine XML produced by the test system.
5. At appropriate, regular and agreed points the NHS CFH CMV lead will provide defect information to the Clinical system supplier either via direct access to MQC, or more likely via reporting from MQC.
6. The Clinical system supplier shall either fix any defects identified, or agree a work-off plan with the CFH CMV lead.

### Outputs

Stand-alone Testing will result in the following outputs:

* Stand-Alone Test Report
* Defects Report

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Confirm that a representative clinical system is available for use |  | I |  | A |  |  | R |
| Arrange access to the clinical system via an appointed clinical system supplier contact |  | AR |  | I |  |  | C |
| Conduct Stand-Alone Testing | I | AR | R[[1]](#footnote-1) | I |  |  | I |
| Produce Stand-Alone Testing Report | I | AR | R | I |  |  | I |
| Produce Defects Report | I | AR | R | I |  |  | I |
| Progress defects | I | C | C | I |  |  | AR |

### Estimated Effort

Completing the Stand-alone Testing task is a function of the number of SCR-relevant modules to be tested, and the degree to which the system has been previously tested. An estimate of elapsed time for Stand-Alone Testing is 2 weeks.

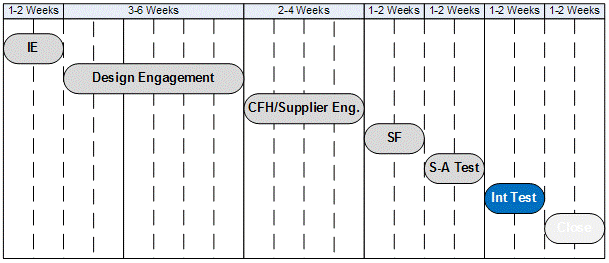
### Tools

Stand-alone Testing will require the following tools:

* Mercury Quality Centre
* Test Report Template (stored in MQC)
* Defect Report Template (stored in MQC)

## Step 6 – Integrated Testing

### Overview Diagram



### Objective

The final testing phase for CMV occurs during Integrated Testing. A representative, compliant version of the clinical system will be installed in the National Integration Centre sandpit environment allowing for full end-to-end messaging with the National Integration Sandpit (NIS). The aim of Integrated Testing is to provide final evidence that the clinical system is clinically safe for SCR messaging, as opposed to the Stand-Alone Testing which aims to catch defects early for resolution.

### Prerequisites

Integrated CMV testing requires the following pre-requisites to be met:

1. A representative version of the clinical system is available in an accessible location.
2. All outstanding defects from Stand-Alone Testing have been fixed in the clinical system, or have been reviewed by the SCR Clinical Safety Officer and accepted with an agreed resolution date.
3. Access to MQC (see section 3.6.8) is available from the testing location.
4. No significant revisions of the clinical system are expected that may invalidate CMV approval of the clinical system.

### Process

Integrated testing will be conducted as follows:

1. The NHS CFH Release Manager will work with the Clinical system supplier to ensure that a representative, stable build is available and connected to the sandpit (NIS) environment.
2. The CFH CMV Lead will arrange access to the test system with the Clinical System Supplier.
3. The CFH CMV Lead will run a series of tests using the Test Scripts documented in MQC. Defects shall be logged using MQC defect management
4. Tools such as an XML viewer and a SNOMED browser shall be used to examine XML produced by the test system.
5. At appropriate, regular and agreed points the NHS CFH CMV lead will provide defect information to the Clinical system supplier either via direct access to MQC, or more likely via reporting from MQC.
6. The Clinical system supplier shall either fix any defects identified, or agree a work-off plan with the CFH CMV lead.

### Outputs

Integrated CMV Testing will result in the following outputs:

* Integrated CMV Test Report
* Defects Report

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Confirm that a representative clinical system is available for use |  | I |  | A |  | C | R |
| Arrange access to the clinical system via an appointed clinical system supplier contact |  | AR |  | I |  | C | C |
| Conduct Integrated CMV Testing | I | AR | R[[2]](#footnote-2) | I |  |  | I |
| Produce Integrated CMV Testing Report | I | AR | R | I |  |  | I |
| Produce Defects Report | I | AR | R | I |  |  | I |
| Progress defects | I | C | C | I |  |  | AR |

### Estimated Effort

Completing the Integrated Testing task is a function of the number of SCR-relevant modules to be tested, and the degree to which the system has been previously tested. An estimate of elapsed time for Integrated CMV Testing is 2 weeks.

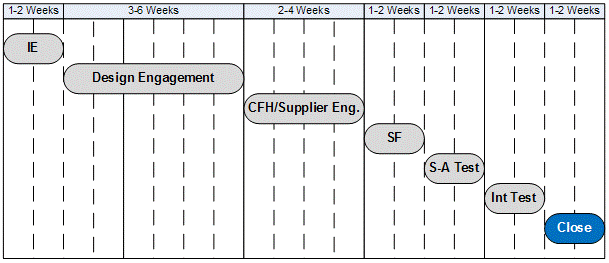
### Tools

Integrated CMV Testing will require the following tools:

* Mercury Quality Centre
* Test Report Template
* Defect Report Template

## Step 7 – Closure

### Overview Diagram



### Objective

Due to the probability that a number for defects or requirements clarifications may be outstanding at the end of CMV testing it is important to undertake a formal project close activity. This shall include:

* Final Defect Report
* Final Requirements Updates Report
* Feedback from key stakeholders
* Final Sign-off of CMV Testing by the SCR Clinical Safety Officer

### Prerequisites

Project Closure requires the following pre-requisites to be met:

1. All testing is complete and signed-off by the SCR Clinical Safety Officer

### Process

Closure will be conducted as follows:

1. The SCR CMV Lead will distribute the final defects report, including work-off plans, to all interested stakeholders
2. The SCR CMV Lead will distribute the final requirements update report, including work-off plans, to all interested stakeholders
3. The SCR CMV Lead shall seek feedback from key stakeholders on the processes, and experience of Clinical Message Validation
4. The SCR CMV Lead shall complete a project closure checklist to ensure that all activities are complete, before seeking the sign-off of the SCR Clinical Safety Officer.

### Outputs

Closure will result in the following outputs:

* Final, published Defect Report
* Final, published Requirements Report
* Summary of Feedback from Key Stakeholders
* Final, completed, signed-off Closure Checklist.

### Roles and Responsibilities

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | SCR Clinical Safety Officer | SCR CMV Lead | Contract Based Clinicians | NHS CFH Release Manager | Clinical Safety Group | National Integration Centre | Clinical System Supplier |
| Distribute final defect report | C | AR | I | I | C |  | C |
| Distribute final requirements report | I | AR | I | C | I |  | C |
| Conduct Feedback from key Stakeholders | C | AR | C | C | C | C | C |
| Complete Closure Checklist | C | AR |  |  |  |  |  |
| Sign-off Closure Checklist | R | AC |  | I |  |  | I |

### Estimated Effort

Completing project closure tasks should take no more than 2 weeks of elapsed time.

### Tools

Closure will require the following tools:

* Closure Checklist
* Final Requirements Report template
* Feedback form/checklist/survey

# Appendix A – Mercury Quality Centre

## Introduction

This appendix is intended to map key concepts in Mercury Quality Centre onto the CMV process. The objective is to allow the CMV team, and any other interested SCR team, to use MQC in a ‘standard’ way, maximising the use of default features.

It is highly recommended that the reader also review MQC tutorials and user guides provided from the MQC user interface.

## Overview

MQC uses a four stage model for managing the quality of software:

* **Requirements** model the fundamental tests that the software must pass.
* **Test Plans** document step-by-step how any one test is executed, and how it maps back to Requirements
* **Test Sets** are groups of many Test Plans run together, with each execution of a Test Set referred to as an instance, or a ‘test run’.
* **Defects** are reported where necessary, optionally linked to a specific test step.

These four main entities are discussed in more detail below, but can be shown in an Entity Relationship Diagram (ERD) as shown in Figure 2.

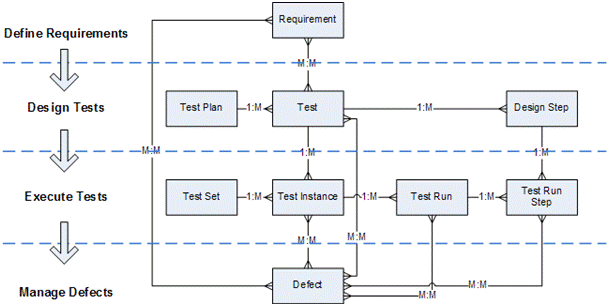


Figure - ERD for the MQC key entities

## Requirements

### Purpose of Requirements

The model on which MQC is based starts with the idea of requirements. The purpose of a requirement within MQC is to guide the test approach i.e. it represents a specific item that requires testing. An example requirement might be ‘The Initial GP Summary core data set shall include all records of allergies and adverse reactions’.

Ideally requirements should map to official CFH requirements e.g. GP Summary Requirements v5.0. However it is possible that requirements ‘lets’, clarifications and other forms of instruction from CFH may also need to be recorded in MQC

In defining requirements for CMV it is considered important to allow for use of the requirements set by other teams, for example Assurance Facilitators. Hence requirements should be structured where possible to not be specific to CMV.

### Relationships

Key relationship notes for the Requirements entity:

* A Requirement can be traced from (covered by) many Tests
* A Requirement can have many Defects linked to it.

### Structuring Requirements

To ensure consistency the structure for requirements shown in Figure 3 shall be adopted:

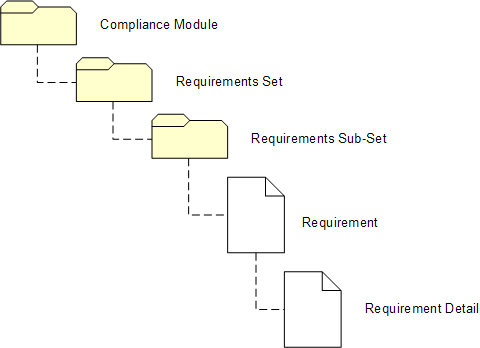


Figure - Suggested organisation of MQC requirements

For example, the requirement to include non-drug allergies in the core contents of a GP Summary message could be represented as shown in Figure 4:

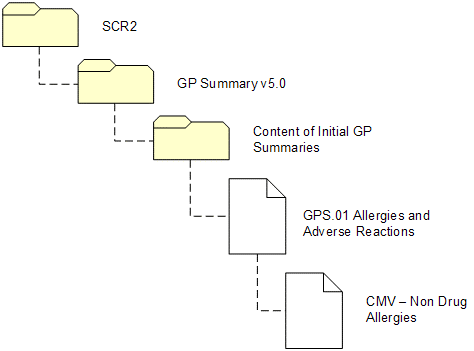


Figure - Example organisation of requirements

In this example, the documented requirement GPS.01 contains the text "allergies to drugs, foods and substances, and drug interactions”. This has been formalised by the CMV team in the Requirement Detail ‘CMV – Non Drug Allergy’. Other requirement details such as ‘Drug Allergies’ will also be added. How this looks in MQC is shown below in Figure 5.

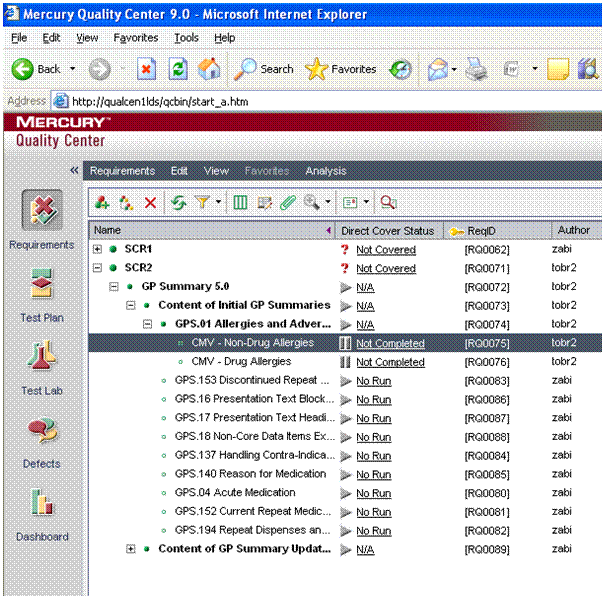


Figure - Example of requirements set up in MQC

## Creating Tests

### Purpose of Tests

Once requirements are defined the next step is to create tests. The MQC test plan consists of many tests (a one-to-many relationship). Each test represents the detailed, system-specific design steps required to exercise requirements. Test Plans will later be executed as ‘Test Runs’ from the ‘Test Lab’ window.

### Relationships

Key relationship notes for the Test entity (Figure 2):

* A Test can trace (cover) many Requirements
* A Test has many Design Steps
* A Test belongs to one and only one Test Plan
* A Test can be used by many Test Instances
* A Test can have many Defects linked to it.

### Test Plan Organisation

Unlike requirements, tests are executed by a specific team for a specific purpose. Hence the organisation is closely related to project work. The suggested structure for tests is shown in Figure 6:

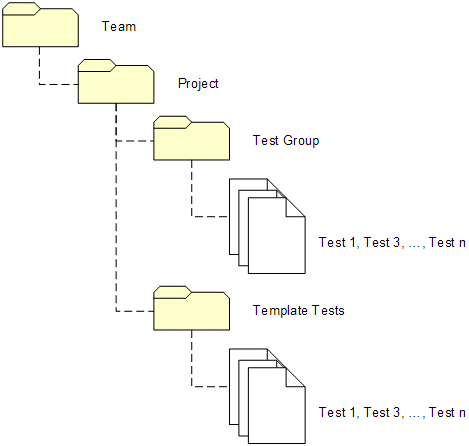


Figure - Suggested organisation of test plans in MQC

Template tests are detailed tests that can be called from other tests. These may be activities or actions that require repetitive execution, such as entering clinical items into a patient’s clinical record.

### Test Granularity

Defining the correct level of granularity for a test is a matter of judgement. If the test is defined as too coarse the number of test steps will become unmanageable. Similarly, if the test is too finely defined then a vast number of tests will need to be executed to cover requirements (see section 4.4.5).

As an example, CMV testing can be broken down as shown in Figure 7:

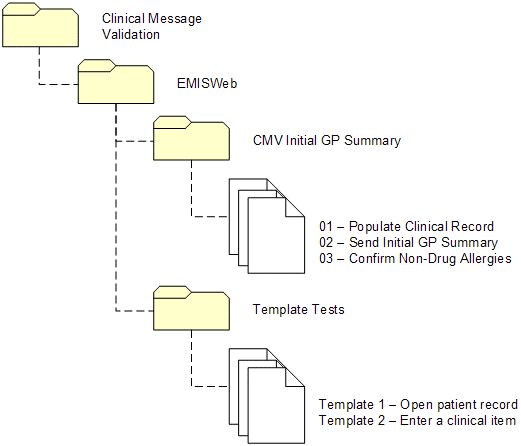


Figure - Example organisation of Test Plans in MQC

Each test in this example is a meaningful, yet manageable set of steps to be executed.

### Use of Parameters

MQC allows parameters to defined and inserted into test steps. These are useful in several ways:

1. When used in a template test, which is in turn called from another test, a value can be assigned to the parameter for use with that test. For example, a ‘Username’ parameter could be defined in a template ‘Logon to system’ test and a different value assigned to ‘username’ in every test that uses ‘Logon to system’.
2. If a parameter is inserted into a test step, but left blank, the user executing the test will be prompted to provide a value at run time. This is useful where a test requires a user to generate test data ‘on-the-fly’, allowing chosen vales to be recorded against each test.

### Requirements Traceability

Each test should be mapped to a requirement. MQC then allows for detailed reporting of how (and eventually, when) each requirement has been tested. Following a test run (see section 4.5) it is possible to provide a detailed report of a clinical system’s level of compliance to a particular set of requirements.

For each test, use the ‘Req Coverage’ tab to link requirements to tests.

As requirements map to an entire test, not the individual test steps, requirements traceability is also a guide to granularity of Test. When a test is failed MQC will mark all associated requirements as failed. Therefore, a Test should address a small set of related requirements that, if the Test fails, can truly be said to also be failed.

For example, consider the following requirements:

* Requirement 1: System shall allow the user to enter a clinical problem using a code
* Requirement 2: System shall send the code to PSIS
* Requirement 3: System shall print off a copy of the code held locally

A test that is too coarse would be as follows:

1. Enter a clinical problem using a code (Req.1)
2. Send the code to PSIS (Req.2)
3. Print off the code held locally (Req.3)

It is perfectly possible for the system to complete steps 1 and 3 successfully, but only fail step 2 (send the code to PSIS). However, recording the test as ‘Failed’ will incorrectly show that neither Requirement 1 nor Requirement 3 has been met when in fact they have.

## Test Lab

### Purpose of the Test Lab

Test Lab is the module of MQC used to execute, and record the success of, Tests that were designed in the above sections.

Tests can be grouped into a ‘Test Set’, which contains many Test Instances. Each test instance is effectively a copy of a corresponding test. A test instance can then be executed any number of times, known as a ‘Test Run’. MQC will record information about how the test run was executed, and its pass/fail status. Test runs also can give rise to defects, which may be linked to the test that has failed.

### Relationships

Key relationship notes for the Test Run entity, and other related entities (Figure 2):

* A Test Set has many Test Instances
* Each Test Instance relates to one and only one Test
* A Test Instance can be have many Test Runs
* Each Test Run has many Test Run Steps, each mapping exactly to the equivalent Design Step pertaining to the Test
* Test Instances, Test Runs and Test Run Steps all can be linked to many Defects

### Organising Test Runs

Test runs are similar to test plans in being related to the project, rather than a generic set of requirements. Test runs are effectively instances of test plans that can be executed repeatedly. An example of how test runs may be organised in MQC is shown in Figure 8. The important point to note is that the same test plan may be pulled into many test runs

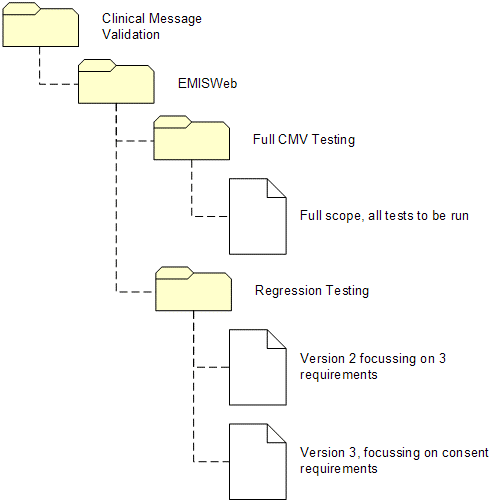


Figure - Example organisation of test runs in MQC.

## Defect Management

### Important Notes about Defects

MQC provides a simple and intuitive interface for managing defects. However, before looking at the Defects module, it is important that all CMV team members understand the definition of a defect:

|  |  |  |
| --- | --- | --- |
| **A Defect is NOT**:  Something that has been proven to be ‘wrong’ with the system. |  | **A Defect IS:**  A test step for which the actual result did not meet the expected result on execution. |

In other words, when the actual test step result does not match the expected test step result, the tester MUST always raise a defect. Remember:

“Log first, ask questions later”

It is a common mistake to investigate an unexpected result to determine if the software is in error. This assumes that only software can be wrong. Other possible outcomes from defect investigations could be:

* The expected result described in the test step is wrong
* The user did not execute the test correctly
* The requirement being tested does not make sense
* Some other technical fault occurred, for example a network outage
* The test data provided by the user breaks certain rules implemented correctly in the software

Clearly, none of the above examples can be blamed on the clinical system itself.

### Defect Resolution Process

Once defects are understood to be, at least initially, an unexpected result to a test step, we need to understand how to tackle the defect. As we have defined ‘defects’ much more broadly than many people usually understand it is not always appropriate to automatically assign the defect to the supplier.

Additionally, other information should be gathered about the defect. For example, where many defects are being raised it is useful to understand the Severity of the defect, i.e. how severe is the impact on the test system?

For effective management of defects we need a Defect Resolution Process that must be followed for all defects. Such processes are usually specific to the organisation, and can become complicated. For CMV testing we have limited resources and scope and therefore a simplified Defect Resolution Process can be used.

The proposed defect management process for CMV is shown in Figure 9.

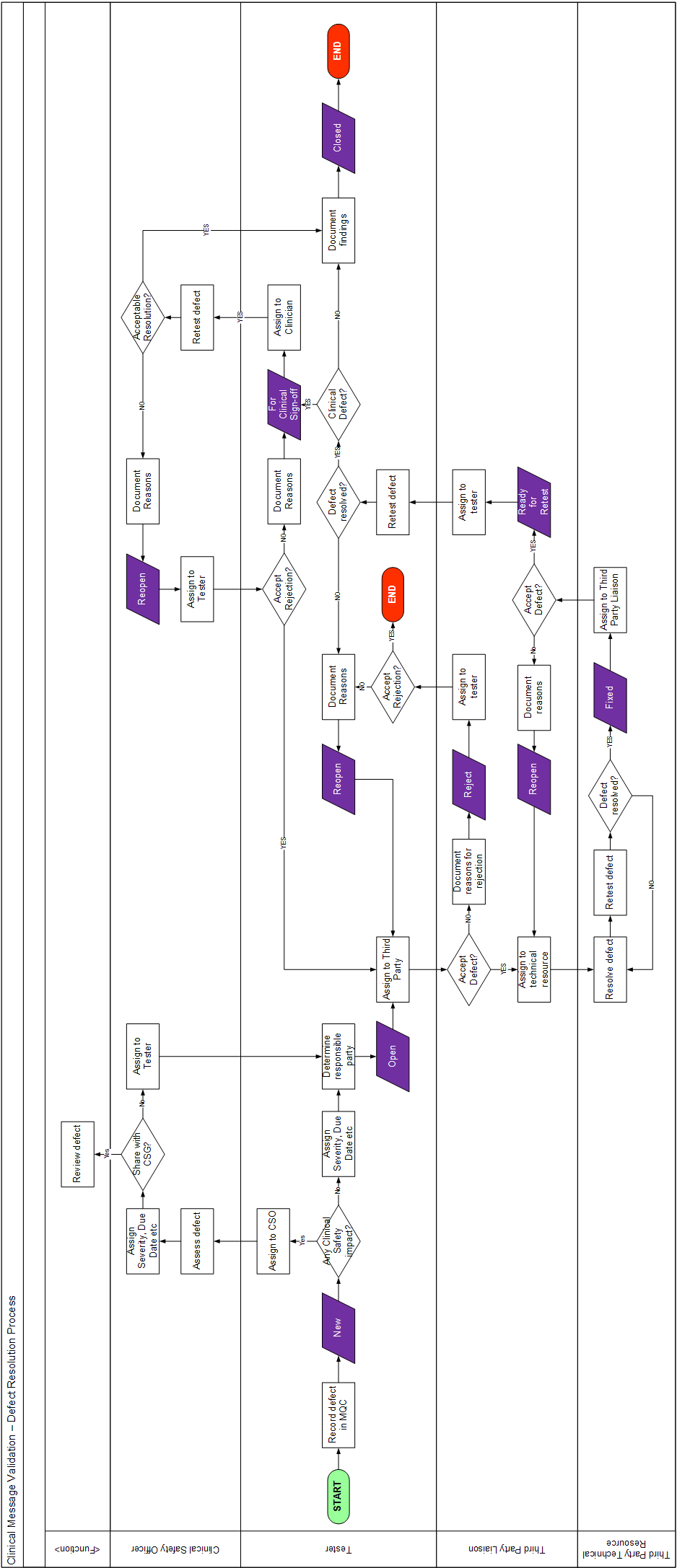


Figure - CMV Team Defect Resolution Process

### Linking Defects

Figure 2 shows a number of relationships for Defects. Specifically:

* A Defect can be linked to many Test Run Test Step
* A Defect can be linked to many Test Runs
* A Defect can be linked to many Test Instances
* A Defect can be linked to many Tests
* A Defect can be linked to many Requirements

These relationships are not exclusive. A defect can be linked to several different types of entity. Defects can also be created in isolation.

Linking defects is valuable as it provides the defect investigator with valuable context information about what was being attempted at the time the defect was noted. Without links, the tester MUST provide detailed context information manually.

MQC does not link defects automatically. The tester must ensure that they always choose to link the defect. If a mistake is made, defects can be manually linked later.

### Accurate Populating of Defects

A defect record is only as good as the information it contains. As a rule, it is not possible to record too much information about the test being executed, and the observed behaviour of the system.

When raising a defect, make use of MQC’s support for taking screenshots, taking dumps of system status, and other attachments.

Whenever an update is made to a defect, the editor should ALWAYS add comments. MQC provides a quick and easy ‘Comments’ function, automatically populating the record with a user/timestamp, under which comments can be added.

**If at any point a defect is not understandable, the assignee is encouraged to return the defect to the assigner for more detail to be provided.**

### Defect Status

Figure 9 also shows the changes of status that a defect goes through. The status is very important in helping everyone to understand what to do with defects.

MQC allows for configuration of status options, however for CMV the Status settings shown in Figure 10 are recommended and have been configured in MQC. Changes of state are shown on the Defect Resolution Process in Figure 9, with a more detailed explanation in Table 1 and Figure 10.

When changing status of a defect it is important that this is only done by the appropriate person as shown in the process flow, Figure 9.

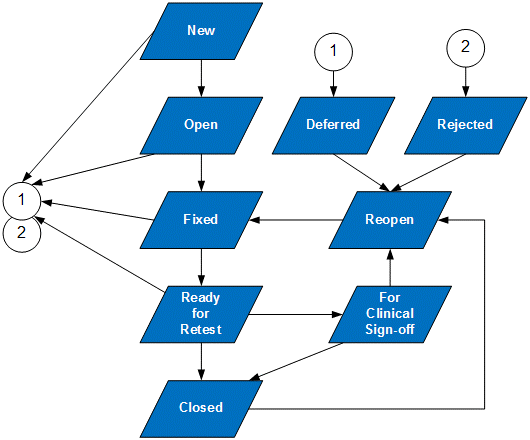


Figure - Status changes recommended for Clinical Message Validation

Table - CMV Status settings for defects

| **Status** | **Description** | **Possible Changes** |
| --- | --- | --- |
| New | The starting status for a defect. Indicates that this defect has only been recorded, but has not been investigated. | * Open * Deferred * Rejected |
| Open | Indicates that the defect has been assigned to a specific individual and should be investigated and resolved. | * Fixed * Deferred * Rejected |
| Fixed | The defect has been investigated, a resolution found, and the resolution has been tested. No further investigation is necessary | * Ready for Retest * Deferred * Rejected |
| Ready for Retest | The defect has a resolution, has been tested, and is now allocated back to the originator for retesting. | * Closed * For Clinical Sign-off * Reopen * Deferred * Rejected |
| Closed | The defect has been resolved and agreed by all parties to be closed. No further action required. | * Reopen |
| For Clinical Sign-of | The defect requires expert clinical review to ensure that the resolution is clinically safe. | * Reopen * Closed * Deferred * Rejected |
| Reopen | Same as ‘Open’, but indicates that this defect has been not been addressed satisfactorily. | * Same as ‘Open’. |
| Deferred | This defect, while valid, does not need to be investigated and resolved at this time. It may be ‘reopened’ at some point in the future.  **A defect may be deferred at any point.** | * Reopen * Rejected |
| Rejected | This defect has been assessed as correct behaviour. For example, user behaviour and incorrect testing may result in a defect being rejected.  **A defect may be rejected at any point.** | * Reopen |

### Summary

The preceding sections are intended to provide some structure for how the full features of MQC – Requirements, Test Plan, Test Lab, and Defects – can be effectively mapped onto the SCR CMV process.

Using MQC in this manner will bring a degree of repeatability and traceability to CMV that has previously relied on the correct storage and exchange of spreadsheets. The desired effect is that CMV testing will require less set up, can be repeated more often, and the quality and speed of documentation will increase.

## Worked Example – CMV Testing of the Initial GP Summary Upload

### Overview

This worked example includes the creation of all required records in MQC including requirements. Of course, once created these elements do not necessarily need to be created again.

### Scenario

The SCR2 Compliance Module includes an updated set of requirements for the GP Summary component, GP Summary v5.0. A new supplier product, “SuperSystem5000” is implementing these requirements and must undergo CMV testing to ensure that the system is clinically safe with respect to SCR messages.

This example considers the execution of tests to confirm that the Initial GP Summary contains all required data as defined in the GP Summary v5.0 requirements.

### Define Requirement

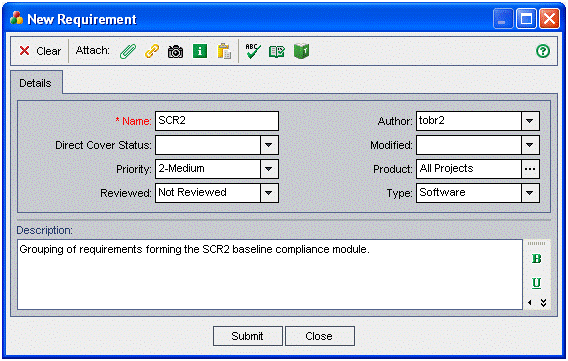
GP Summary v5.0 defines a requirement GPS.01 as follows:

The system MUST include the complete record of patient allergies and adverse reactions, including allergies to drugs, foods and substances, and drug interactions.

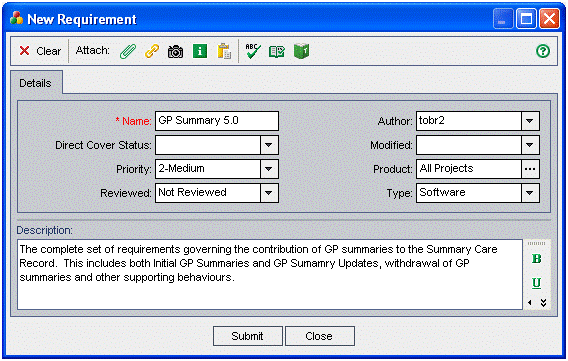
Refer to "NPFIT-EP-DB-0007.05 Representation in EPR of Allergic Reactions, ADRs and Intolerance to Pharmaceutical Products v1.5". This document is part of the Summary Care Record Baseline.

Assuming this requirement does not already exist, the tester must create it as the item to be tested. Using the recommended organisation of requirements defined in section 4.3.3, we create the following items:

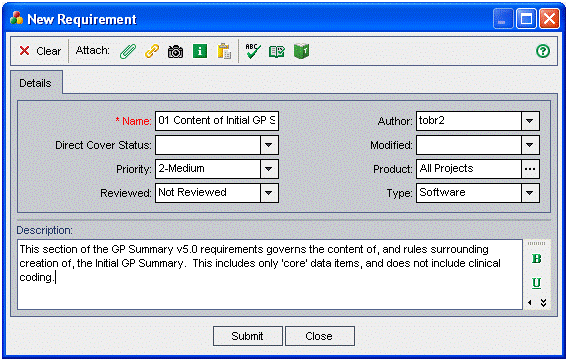
1. Parent Requirement for the ‘SCR2 Compliance Module’:



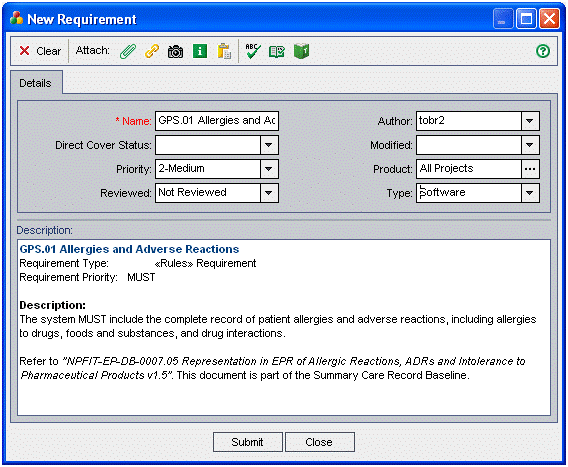
1. A child requirement to ‘SCR2’ for the GP Summary v5.0 requirements:



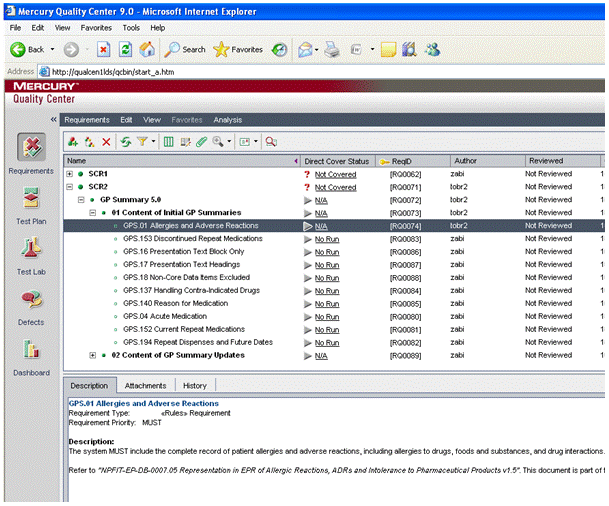
1. A child requirement to ‘GP Summary v5.0’ for the sub-section of requirements, “01 – Contents of the Initial GP Summary”:



1. A child requirement to “…Content…” for the actual GPS01 requirement:



The resulting arrangement of requirements now looks as follows:

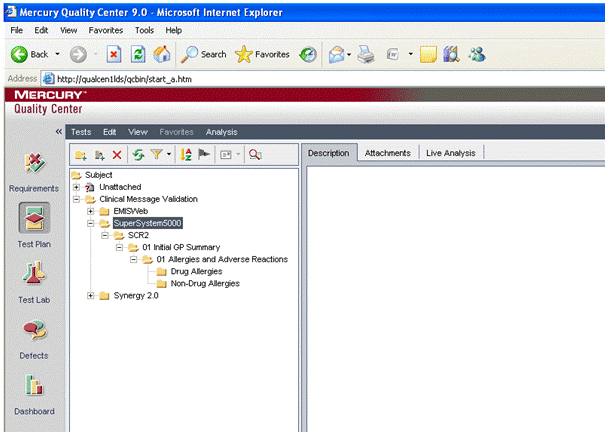


### Define Tests

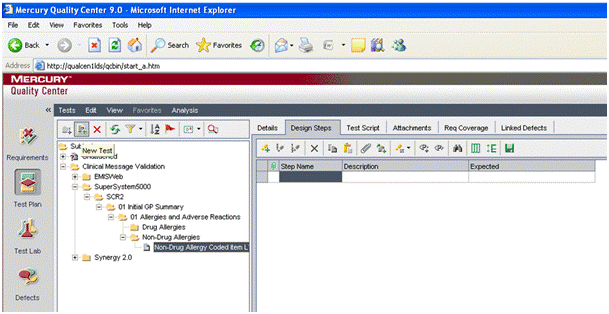
Having defined the ‘what’ that is to be tested, the second step is to define the ‘how’ i.e. to generate a Test.

Switching to the ‘Test Plan’ module, and using the arrangement of Tests suggested in section 4.4.3:

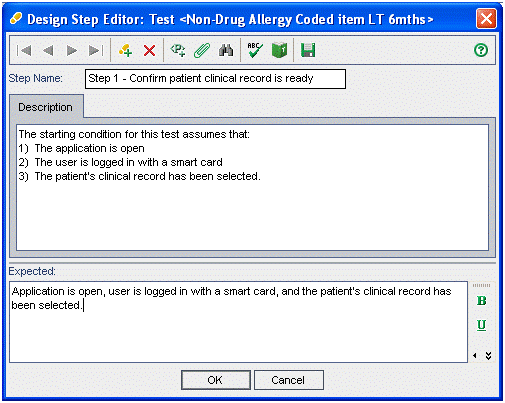
1. Create a top-level Test Folder called ‘Clinical Message Validation’ (this allows other teams to define their own sets of tests).
2. Tests are ‘how’ items, so are specific to the supplier. Create a sub-folder to ‘Clinical Message Validation’ for the project in question, in this case “SuperSystem5000”.
3. We are testing this system against SCR2 requirements. However the system could equally be tested against other future requirements, so we create an ‘SCR2’ sub-folder to keep conflicting Tests separate.
4. We will test the Initial GP Summary messages as a separate set of tests. Therefore another sub-folder is created.
5. The CMV team is interested in testing two types of allergies; ‘Drug Allergies’ and ‘Non-Drug Allergies’. Each type of allergy will be tested in numerous ways, hence two more sub-folders are created.
6. The resulting hierarchy of test folders is as follows:



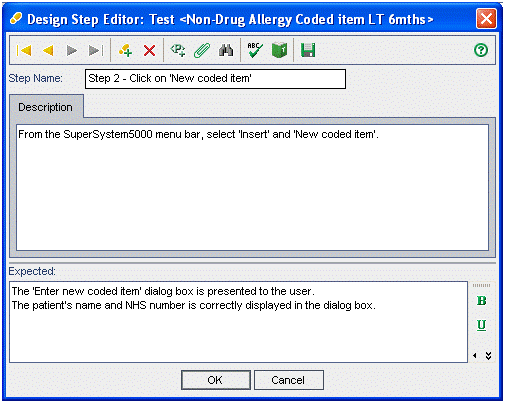
1. Now we define the specific tests. We will define specific tests beginning with a non-drug allergy, entered as a coded item within the last 6 months:
   1. While having ‘Non-Drug Allergies’ highlighted, create the Test by clicking on ‘New Test’



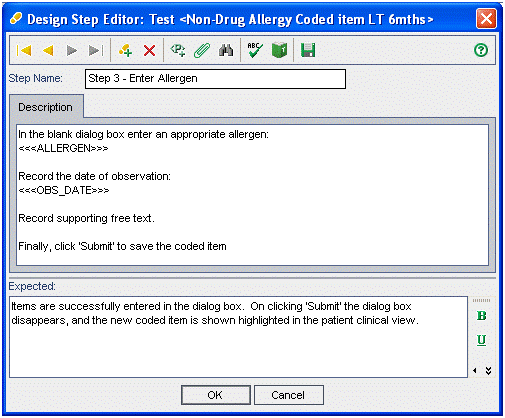
* 1. This test will assume that the application is already open, and the patient clinical record selected. This assumption is documented as a first ‘Detail’ step. To create this step, the user clicks on ‘New Step’ and enters the following information:



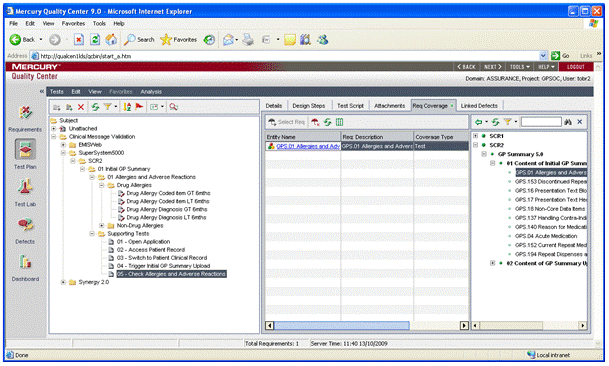
* 1. Create a step to open the ‘new coded item’ dialog box. Note that this is very specific to SuperSystem5000:



* 1. The user now creates a step to record the actual allergy, and actual date of the item. This uses parameters for “ALLERGEN” and “DATE” allowing the tester to record these values at execution (see section 4.4.5):



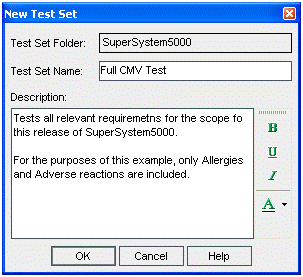
1. We will create further, similar tests as follows:
   1. Non-Drug Allergy, entered as a diagnosis, more than 6 months ago:
   2. Non-Drug Allergy, entered as a coded entry, more than 6 months ago:
   3. Drug Allergy, entered as a diagnosis, within the last 6 months:
   4. Drug Allergy, entered as a coded entry, within the last 6 months:
   5. Drug Allergy, entered as a diagnosis, more than 6 months ago:
   6. Drug Allergy, entered as a coded entry, more than 6 months ago:
2. Generic tests are added for:
   1. Opening the application
   2. Selecting the patient
   3. Accessing the patient’s clinical recod
3. The primary purpose of these tests however are to ensure that the clinical content is correctly transmitted in an Initial GP Summary upload. This requires a “Trigger Initial GP Summary Upload test” to be created.
4. Finally, to actually check the clinical content a “Check Allergies and Adverse Reactions” test is created. Note that this Test is maped to the requirements created above using the ‘Req Coverage’ tab:



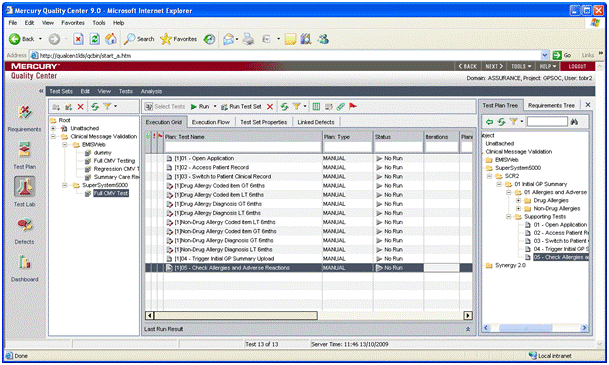
### Generate Test Run

Having created a test plan, at a later point the CMV tester is now ready to execute a test. This requires first creating a test run.

1. The suggested organisation (see section 4.5.3) follows that of the test plan. Create a high-level folder for “SuperSystem5000”.
2. A test set is created to hold all the tests that will form part of a “Full CMV Test”.



1. Individual Tests are now added to the Test Set as Test Instances. Note that we user the generic ‘Open application’ Tests, then add in ‘Allergy’ Tests, then finally add the ‘Trigger Upload’ and ‘Check Allergy’ Tests:

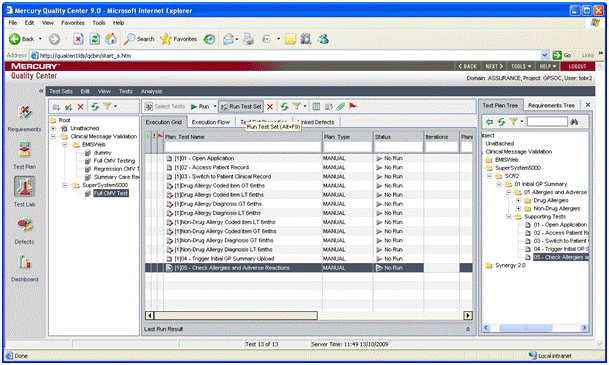


### Execute Test Run

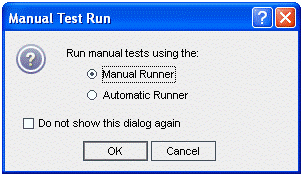
Once a Test Set has been created in Test Lab the set is ready for execution by the tester. Each Test Set can be executed any number of times, and the results recorded in MQC.

A typical execution flow may be as follows:

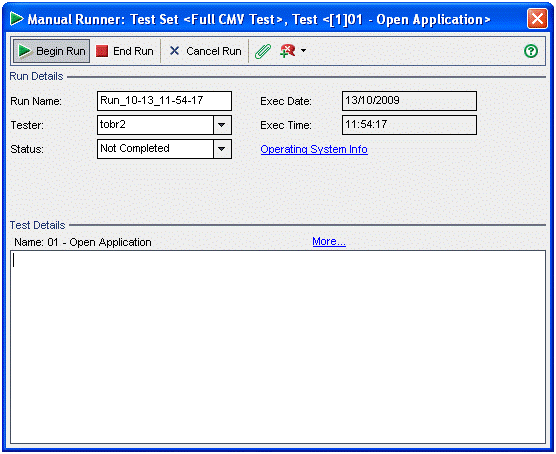
1. Select the Test Set to be run (Full CMV Test) and click on ‘Run Test Set’:



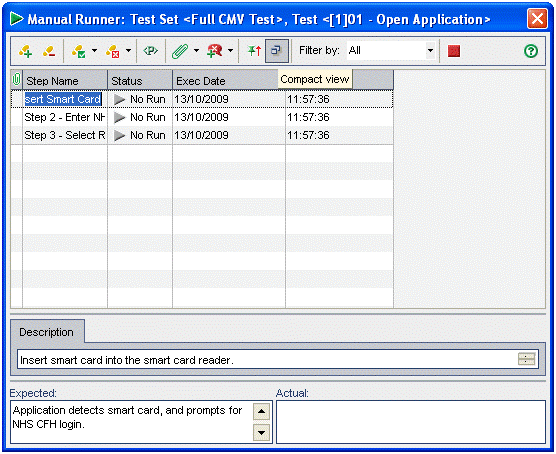
1. The test execution windown opens. Select ‘Manual’ and click ‘OK’.



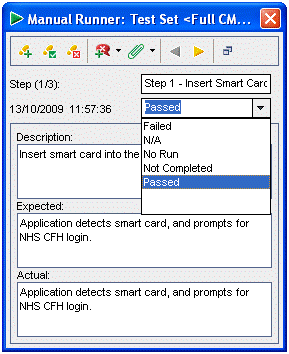
1. The Manual Runner is now shown. Click ‘Begin Run’.



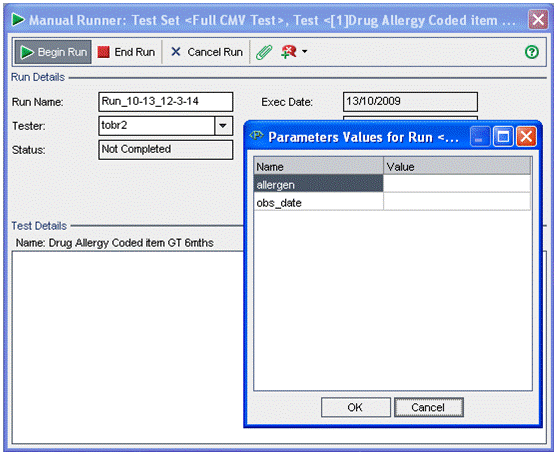
1. The Manual Runner window shows the first test instance to be run, in this case ‘Open Application’. For convenience it is often best to switch to compact view:



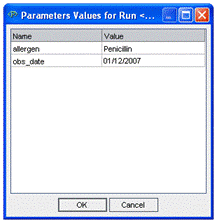
1. The compact view is better for viewing instructions, expected results, and actual results. In this case, the actual results match the expected results, and the test step instance has passed. This is recorded using either the status drop-down, or the tool bar button as shown:



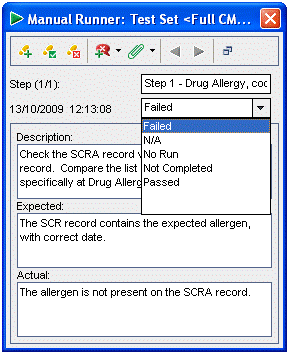
1. The test continues to be executed without failure. When the first allergy-entry test is reached, MQC prompts for the parameters defined in the Test:



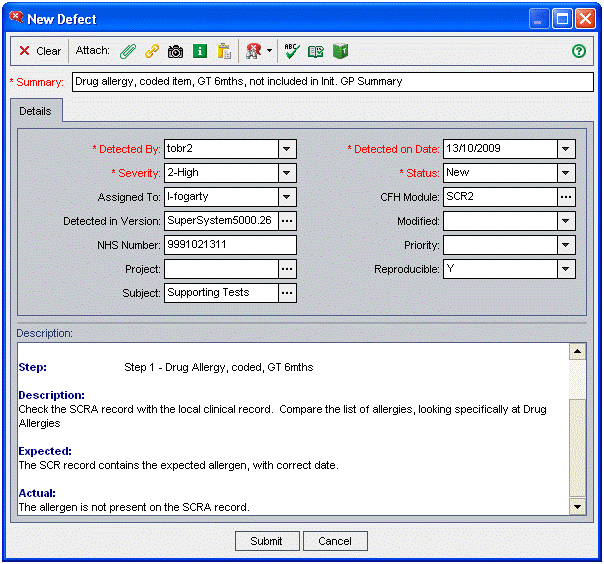
1. The tester enters the parameters being used and clicks okay. The use of parameters enables the tester to use a list of possible clinical terms to ensure variation of testing:



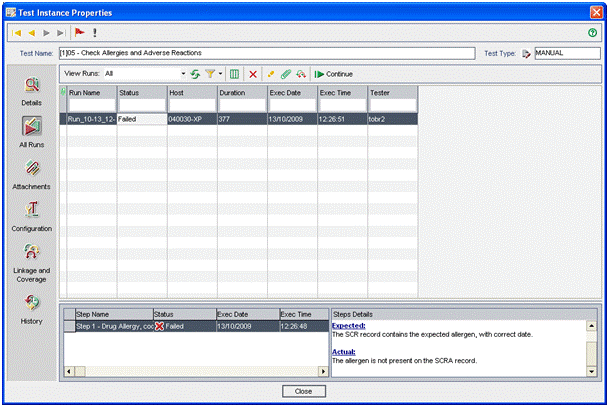
1. The tester continues the test run. The final tests involve comparing the content of the Summary Care Record Application (SCRA) with the local clinical record. In our example, the tester notes that the “Drug allergy, coded item, entered greater than 6 months ago” is absent. The tester records this in the actual results, and marks the test as ‘FAILED’:



1. Before moving to the next test, the tester also raises a defect against this test step by clicking on the defect button (shown in the image above). The defect is recorded in detail to allow future investigation. Screenshots are also included to verify the defect finding.



1. The remaining tests are executed, any other defects raised, and the test run ended. The status of the test run including its individual executions can be seen in the Test Lab:

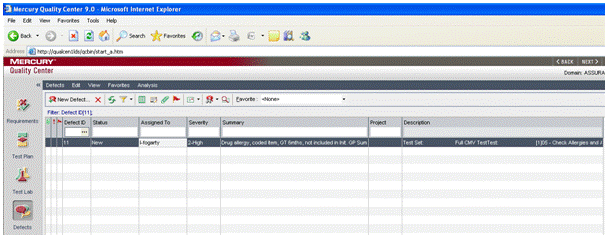


### Manage Defects

At the conclusion of testing, a number of defects are likely to have been created. It is important to note that a defect should be raised whenever an actual test step result does not match the expected result. It is poor practice to investigate the defect first to determine if it is a software bug, before recording the defect in MQC.

Defect management is detailed in section 4.6, but some examples for managing the defect raised in this worked example are provided below.

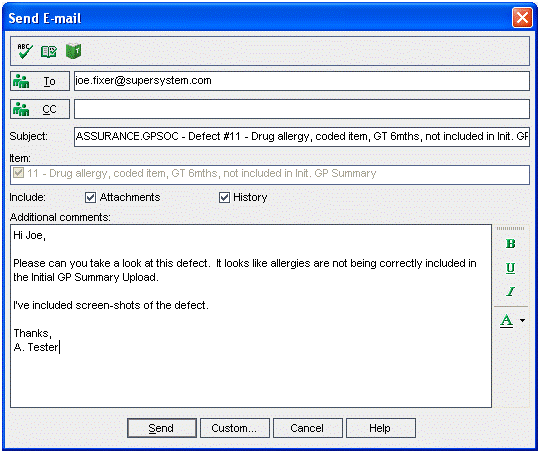
The defect module screen is shown below:



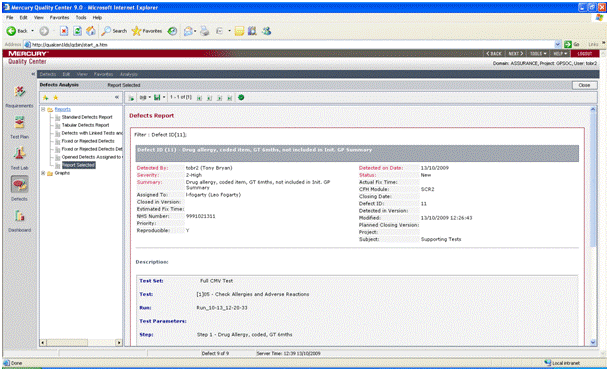
Analysis Function

Email Button

1. The defect may, in the first instance be emailed to another party for investigation:



1. The tester can also generate a report of all defects relevant to SuperSystem5000 using the analysis function. Reports can be printed, or saved as HTML documents and sent to third parties:



1. Finally, the defect list can be exported as Excel, Word or HTML files. This required the tester to define what columns to be included. Example export files are embedded:



Regardless of the exact mechanism used to exchange defect information with the supplier, the CMV team will most likely be responsible for updating the MQC record as third parties provide responses.

Please refer to the Defect Management process for a more detail treatment of how to handle defects.

# Appendix B – Planning Guidelines

## Objectives

In managing the CMV process there are many dependencies on other activities and teams. A detailed planning process is therefore required to ensure the timely allocation of resource, and early identification of clashes.

The process laid out in this document provides a structure for CMV work which allows an estimate of the work required to be generated. This in turn allows long-range planning. This section provides supporting information for the CMV planning process.

## Planning Assumptions

The following assumptions are made regarding the planning process:

1. The CMV Lead role is available 4 days per week for CMV work. The remaining 1 day per week is nominally assigned to non-specific tasks including:
   * Administration, reporting and planning
   * Defect management and Requirements liaison
   * Sickness, vacation, training time not already planned in.
2. Contract-based CMV-trained clinicians are available at about 6 weeks notice

## High-Level Demand Management Process

The demand on the CMV team, and supporting teams such as the NIC, will be managed as follows:

1. During Step 1 – Initial Engagement (see section 3.2) the CMV Lead will meet with appropriate release maangers to determine high-level scope and timelines for CMV work.
2. The agreed high-level scope shall be used with a standard estimating model to provide approximate effort for the CMV task.
3. An updated Capacity Plan will be published by the CMV lead, and discussed regularly with each release manager.
4. The CMV Lead will work with release managers and project management to ensure that clashes and risks are adequately minimised.
5. At each subsequent step in the standard process the capacity plan shall be reviewed for accuracy, and estimates updated.

**Important: It shall be the responsibility of the release manager to notify the CMV Lead should the agreed scope and dates change**

## Estimating Model

### Purpose

A standard estimating model has been devised to ensure that a consistent long-range planning approach is taken. At Step 1 – Initial Engagement (section 3.2), scope will be agreed with the release manager concerned which will in turn be fed into the estimating model to provide an initial estimate for effort and time required.

### Estimating Model Approximations

An estimating model is intended to provide an approximate view of effort required, the results of which should be continuously reviewed as more information becomes available. The following approximations/methods are built into the estimating model:

1. Estimates are made for the work required for each of the seven process steps outlined above individually.
2. SCR CMV compliance work is broken down into modules. Each module represents a discrete set of tests that may be applied, such as ‘GP Summary v5 Compliance’ or ‘CDA Compliance Module’.
3. The model is therefore based on the amount of work *per module* required to complete *each step*.
4. It is possible that a supplier is implementing many modules at once, which will drive up the effort required. For example, a supplier may be implementing:
   * GP Summary version 5 (send only)
   * GP2GP (will require SCR-GP2GP interoperatibility testing)
   * CDA Accident & Emergency Messages (read only)
   * CDA Out of Hours Messages (send only)

This constitutes four modules, which will therefore require more effort than a system restricted to only ‘GP Summary v5 send only).

1. Each step also carries an overhead, i.e. a a minimum level of effort required regardless of the number of modules being assured.

### Location and Maintenance

The estimating model is a living document, to be revised over time as more experience with the 7-step process is obtained. Hence the estimating model is not included in this document, but can be found at:

[https://www.portal.nss.cfh.nhs.uk/sites/crs/scr/DAT/Shared%20Documents/11%20Clinical%20Message%20Validation/02%20-%20Team%20Management/03%20-%20Estimating%20Model/CMV\_EstimatingModel\_VersionFinal.xls](https://www.portal.nss.cfh.nhs.uk/sites/crs/scr/DAT/Shared%20Documents/11%20Clinical%20Message%20Validation%20and%20GP%20Summary%20Documentations/02%20-%20Team%20Management/03%20-%20Estimating%20Model/CMV_EstimatingModel_VersionFinal.xls)

1. Contract-based clinicians will be used where necessary to execute testing, depending on availability of CFH resource. [↑](#footnote-ref-1)
2. Contract-based clinicians will be used where necessary to execute testing, depending on availability of CFH resource. [↑](#footnote-ref-2)