SCM testing procedures

Should Cost Modelling



MAY 2021

Version 1.0

**Contents**

[1. Context 2](#_Toc64534209)

[2. Introduction 3](#_Toc64534210)

[3. Testing Principles 4](#_Toc64534211)

[4. Testing Process 12](#_Toc64534212)

[5. Verification 15](#_Toc64534213)

[6. Recomputation/Parallel Build 26](#_Toc64534214)

[7. Validation 28](#_Toc64534215)

[8. Other Testing 34](#_Toc64534216)

[Appendix 1 – QA Report Template 36](#_Toc64534217)

[Appendix 2 – Test Memo Template 41](#_Toc64534218)

# Context

* 1. The requirement to produce a Should Cost Model (SCM) when making sourcing decisions and contracting outside suppliers for the delivery of **public services** is set out within the [Sourcing Playbook](https://www.gov.uk/government/publications/the-outsourcing-playbook) (see Chapter 3) and for **public works projects or programmes** within the [Construction Playbook](https://www.gov.uk/government/publications/the-construction-playbook) (see Chapter 5).
  2. The Sourcing and Construction Playbooks set out when contracting authorities should produce an SCM, which functions are responsible for them (see Ownership, Knowledge, Understanding and Awareness framework), and how SCMs fit within the procurement lifecycle. The following additional guidance is also available on SCMs:
* [SCM Guidance Note](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/891146/Should_Cost_Modelling_-_Guidance_Note_-_June_2020.pdf) - outlines what SCMs are, when and why contracting authorities should produce them, and key considerations around developing and/or procuring them;
* SCM Development Guidance - provides contracting authorities with guidance on using internal resources to design, develop, manage, test and govern SCMs; and
* SCM Technical Build Guidance - guidance, based on good practice principles for building SCMs. It is technical in nature and aimed at people who will be building SCMs.
  1. Practitioners should also consult existing good practice guidance including HM Treasury’s [Macpherson](https://www.gov.uk/government/publications/review-of-quality-assurance-of-government-models) report, [Aqua Book](https://www.gov.uk/government/publications/the-aqua-book-guidance-on-producing-quality-analysis-for-government) and [Green Book](https://www.gov.uk/government/publications/the-green-book-appraisal-and-evaluation-in-central-governent).
  2. This SCM Testing procedures document, which is one of a number of tools and templates produced by the Sourcing Programme to support the production of SCMs, draws on the above publications and other, more general, government guidance.
  3. The guidance herein aligns with existing guidance within some departments. It should be applied in a manner that is proportional to the risks associated with a specific SCM and its use. Whilst adherence to it is not mandated, it is recommended where there is nothing similar in use within the contracting authority.
  4. You should consult the Cabinet Office Sourcing Programme for further information or before planning an SCM for complex services, projects or programmes via [sourcing.programme@cabinetoffice.gov.uk](mailto:sourcing.programme@cabinetoffice.gov.uk).

# Introduction

Testing a model is not a one person, 30-minute, tick box exercise. It is an important activity intended to rigorously challenge a model and is fundamental to producing an SCM that provides appropriate, high quality, insight.

This document is relevant to anyone wanting to understand the model review process, testing good practice or details of specific test procedures. It is equally applicable to model developers, Model SRO’s or practitioners performing testing.

This document details testing principles, the testing process and some key test procedures that can be applied when testing a model.

In many cases, the majority of testing will take place towards the end of the model development lifecycle. However, in addition to testing, Quality Assurance (QA) should take place throughout the model development lifecycle (see Figure 1). QA requirements should be considered at the model planning stage and documented in an QA Plan for approval.

***Figure 1: Model Development Lifecycle***



*\*Key documentation to produce included in square brackets ‘[ ]’ are optional, depending on requirements.*

# Testing Principles

This section sets outs the general testing principles to be followed when testing a model. It covers the following areas:

* Quality Assurance (QA) and testing;
* Roles and Responsibilities and the Principle of Independence
* Basis for Comparison;
* Documentation;
* Query Logs;
* File Management and Version Control;
* Testing Folder Structure;
* Determining what testing to apply; and
* Incremental testing.

## Quality Assurance (QA) and testing

### QA is predominantly centred on helping to ensure that a model is developed to a sufficiently high standard. It refers to the governance, control, review and test procedures in relation to a model. Testing is an important part of QA and includes:

### Checking the model for errors and alignment with the model Specification (Verification); and

### Checking that the model is appropriate, i.e. fit for the purpose for which it is being used (Validation).

### Whilst, testing is only a subset of the full suite of QA that should be performed over a model, it is very important. See [Aqua Book](https://www.gov.uk/government/publications/the-aqua-book-guidance-on-producing-quality-analysis-for-government) for further details.



## Roles and Responsibilities and the Principle of Independence

### The model development process will invariably involve several different roles and stakeholders, typically including some or all of the following:

|  |  |
| --- | --- |
| Model SRO | Quality Assurers\* |
| Model Architect | Model Operator |
| Model Developer | Model Customers |
| Data Providers |  |

### Further details on Roles & Responsibilities can be found in the SCM Development Guidance.

\*Quality Assurer is a general term used to describe those who undertake quality assurance. In the context of this document, which focusses on model testing, the term Quality Assurer refers to the individual(s) undertaking model testing.

### Fundamental to testing is the **Principle of Independence**. Regardless of the test procedure being performed, it is critical that the quality assurer is independent of the model developer and, ideally, independent of other stakeholders associated with model development.

### Quality assurers should be suitably qualified and experienced to discharge their responsibilities. The [Review of Quality Assurance of Government Models](https://www.gov.uk/government/publications/review-of-quality-assurance-of-government-models) notes that there is no substitute for expertise and experience. The effective performance of testing is essential and it should not be viewed as a task for junior resources alone.

## Basis for Comparison

### When testing a model, it is important to have a **Basis for Comparison** to test the model against. For example, if performing Verification, the quality assurer would need to know what the logic of the model should be (as documented in the model Specification) in order to effectively test the model. Similarly, when performing an Analytical Review, the review should be performed against documented expectations of the results of each test.

### Without testing ‘against’ a Basis for Comparison, there is a risk that testing will not be effective. Further details on the Basis for Comparison for each type of test can be found in the detailed test procedure chapters below.

## Documentation

### Documenting what testing has been performed, when it was performed, who undertook it and the results of the testing is an important activity. It provides an audit trail and supports communication of testing activities to key stakeholders, including the Model SRO as part of the model sign off process. Without evidence of a sufficient and robust testing process a model should not be signed off for use. This will involve setting out what testing has been performed on the model, along with the findings from the tests and clearly documenting any outstanding or unresolved issues.

### Testing should be clearly documented and in sufficient detail to enable reperformance, if necessary, by an independent person. Clarity over what testing has been performed and the results of that testing, including any outstanding issues, helps to:

### Ensure that there is a full audit trail and provides a mechanism by which QA and testing can be signed off;

### Support further or incremental testing by clearly documenting what and how prior testing has been performed; and

### Support third party scrutiny of the model development process and the steps taken to help ensure that a model is developed to a sufficiently high standard;

### Support the model being signed off for use by the Model SRO.

### Notably, a model that has been through QA and testing does not necessarily mean that it is appropriate to use. There are a number of factors that should be considered when signing off a model for use. For example:

### Models are built for a specific purpose and if requirements change or if a model is used for something other than what it was originally designed for, it may not be fit for purpose;

### The model may have limitations, including those identified during or relating to QA and testing (e.g. the scope of QA and testing), that constrain how it should be used or what it should be used for; and

### Models may be modified after having been subjected to formal QA and testing. These modifications include both structural alterations or additions, and updates or changes to the model’s data or assumptions.

**QA Plan**

### During the Plan phase of the model development lifecycle (see Figure 1***: Model Development Lifecycle***), the overall QA, including test procedures, expected to be performed on the model should be set out in a QA Plan. The QA Plan should be reviewed and signed off as part of the Plan phase. The QA required throughout the model development lifecycle may change and, as such, the QA Plan should be reviewed periodically and updated accordingly.

### Included within the QA Plan should be a Test Plan, which sets out the specific test procedures that will be performed on the model.

### There is no prescribed format for a QA Plan but it is suggested that it covers the following:

* Name, Job Title who prepared QA Plan
* Date of Preparation
* Name, Job Title who reviewed & approved QA Plan
* Date of Approval of the QA Plan
* Project Background – key information about the project
* Model Background – key information about the model, who is building it, timescales etc.
* Details of the Model Development Team
* Details of the planned Model Development Process including model governance/control
* Model/project risks and planned mitigations
* Test Plan – Planned Test procedures detailing:
  + Test procedures to be performed
  + Risk(s) they are intended to mitigate
  + Resourcing requirements
* QA Limitations

This is not an exhaustive list and is only a guideline of the key information that should be included in a QA Plan. An SCM QA Plan Template is available.

**QA Report**

### During the Test phase of the model development lifecycle (see Figure 1***: Model Development Lifecycle***), test procedures will be performed on the model. A QA Report, aligned to the QA Plan, should be produced to show what QA and testing has been performed and any key findings, including any deviations from the original QA Plan. It is suggested that a QA Report covers the following:

* Name, Job Title who prepared QA Report
* Date of Preparation
* Name, Job Title who reviewed & approved QA Report
* Date of Approval of the QA Report
* Project Background – key information about the project
* Model Background – key information about the model, who built it, timescales etc.
* Details of the Model Development Team
* Details of the Model Development Process including model governance/control
* Model/project risks and mitigations
* Test procedures performed on the model:
  + Test procedures performed
  + Risk(s) they are intended to mitigate
  + Details of model versions that each procedure was performed on
  + Supporting evidence for each procedure (see Section details on Test Memos below)
* Conclusion – this should note whether any important issues (raised through the review process) remain outstanding and, taking these and the scope/ limitations of QA into account, whether the model is recommended for release. Any key issues from underlying Test Memos should be included here.

This is not an exhaustive list and is only a guideline of the key information that should be included in a QA Report. A QA Report Template can be found in Appendix 1 – QA Report Template.

An Excel version of the QA Report Template could also be included in the SCM within a ‘QA\_Report’ tab.

**Test Memos**

### Each test procedure should be documented to provide clear evidence of the testing that has been performed. This would usually be in the form of a Test Memo, although it could be set out in an email.

### Testing should be documented in sufficient detail to enable reperformance, if necessary, by an independent person.

### A model is likely to undergo several different test procedures that may be performed by different quality assurers. As such, it is likely that there will be multiple Test Memos to support the QA Report, although these could be combined. For example, a Test Memo prepared by one quality assurer to cover Verification and a separate Test Memo to cover Analytical Review prepared by a different quality assurer.

### There is no prescribed format for a Test Memo, however it is suggested that it covers the following:

* Name, Job Title who prepared Test memo
* Date of Preparation
* Project Background – key information about the project
* Model Background – key information about the model, who built it, timescales etc
* Review Cycles completed including:
  + Model version details (Filename, Version, Date, File Size)
  + Specific tests performed
  + Findings from Review Cycle
  + Details of Queries Raised (documented in a Query Log – see Section 3.5)
* Any incremental testing performed (see Section 3.9 for further details)
* Conclusion – this should make clear if there are any important issues (raised through the review process) that remain outstanding or if model testing is complete with no outstanding issues.

This is not an exhaustive list and is only a guideline of the key information that should be included in a Test memo. A Test Memo Template can be found in Appendix 2 – Test Memo Template.

### It is usually the responsibility of the model developer to coordinate the testing and documentation, but this may vary from project to project. Regardless, it is essential that the Model SRO is ultimately satisfied with the QA and testing performed on the model.

## Query Logs

### Query Logs should be completed for all testing performed on a model. They record and track the history of queries raised by the quality assurer in relation to potential issues with the model and/or its associated documentation. They help to ensure that they are appropriately investigated and resolved by the model developer and, thereafter, that that changes are checked and closed by the quality assurer.

### Query Logs should be sufficiently detailed so that the model developer can understand what the potential issue is and, in-turn, what action may be required. There is no set format for a Query Log, but suggested areas for inclusion are:

### Query Reference Number

### Query Type (e.g. Category 1/2/3/Clarification – see below)

### Model Filename, Version and Date

### Worksheet Name

### Cell Range (e.g. AA12:CO16)

### Label (e.g. Inflation Index)

### Formula (Top left cell only if a repeatable calculation block)

### Query (e.g. "The formula is not consistent across the full timeline")

### Model Developer Response (to support the checking of changes by the quality assurer and enable queries to be closed, e.g. "Formula updated to be consistent across the timeline")

### Query Status (Open/Closed)

### The Table below lists indicative Query Type classifications:

|  |  |
| --- | --- |
| Query Type | Description |
| Category 1 | Actual Error within the model (e.g. incorrect formula) |
| Category 2 | Potential Error in the model (e.g. if inputs change) |
| Category 3 | Good Practice Point (e.g. falls short of good practice) |
| Clarification | Quality assurer clarification (e.g. need more information) |

### Whilst the Query Log should clearly articulate the quality assurer’s concern it must not suggest any form of solution for the model developer to implement as this could effectively then lead to self-review.

## File Management and Version Control

### File Management and Version Control is critically important, both during and after testing.

### The model developer will always hold the master version of the model up until its formal release. Once the model is submitted for testing the model developer should not make any changes to the model until the quality assurer has completed their ‘review cycle’. This will allow for an audit trail to be maintained.

### If changes are required following a ‘review cycle’ they should be minimised and only made in relation to queries raised by the quality assurer. It should not be seen as an opportunity to ‘tweak’ other areas of the model. It is important to limit, as far as possible, structural changes such as adding and, in particular, deleting columns or rows, as this can substantially increase the time required by the quality assurer to check the changes.

### Any changes should be made in a new version of the model (increased version number) and a model Version Control Log should be updated with details of all the changes made.

### It is likely that several ‘review cycles’ will be required for each test procedure. This is where queries raised by the quality assurer are rectified by the model developer and the updated model is passed back to the quality assurer to check whether queries can be closed. The number of review cycles is driven by both the extent to which queries are resolved and the (potential) introduction of new issues as a result of changes to the model. Being able to trace the history of queries and changes through the review cycle requires robust approaches to File Management and Version Control.

### It is likely that several different test procedures will be required for each model and this further increases the importance of robust approaches to File Management and Version Control. For example, the following sequence of events may occur:

### Version 046 of the model is provided to a quality assurer to perform Verification;

### Queries from Verification are resolved by the model developer in version 047 of the model and checked and signed off as closed by the quality assurer;

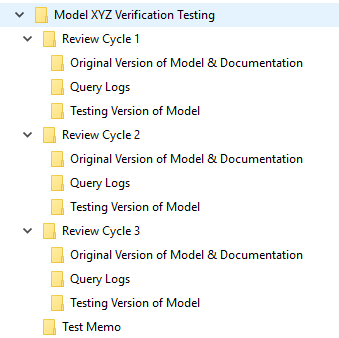
### Version 047 is then is then provided to a different quality assurer to perform an Analytical Review; and

### The queries from the Analytical Review are resolved and updated in version 048 of the model and checked and signed off as closed by the quality assurer.

### As such, it is important that each version of the model is maintained for traceability and a clear record is maintained of what testing was performed and what changes were made on what version of the model. Further details of documentation can be found in Section 3.4.

## Testing Folder Structure

### To support File Management and Version Control it is helpful to set up a logical and consistent folder structure to save the files associated with each test procedure (e.g. Model XYZ Verification Testing). There is no prescribed format, however, the following is an indicative folder structure for the quality assurer:



## Determining what testing to apply

### There is no one size fits all rule for testing a model and the measures taken should be proportionate to the significance or criticality of the decision and the risk presented by the model. More complex or sophisticated models generally have a higher likelihood of error and may require the application of more advanced or rigorous testing. Ultimately, the testing performed on a model will be guided by the level of certainty required over the model and its output and the level of risk tolerance towards errors that is acceptable.

### The IMA Tool can be used to inform the selection of appropriate QA and testing. This is however a guide and provides an indicative set of test procedures to apply. Consider, on a case-by-case basis as part of QA planning, whether more stringent QA and testing is appropriate.

### Ultimately the extent of testing performed on the model will need to be agreed and signed off by the Model SRO (or other appropriate governance). The test procedures to be applied to a model should be determined during model planning (see Section 3.4.3).

### When determining which test procedures should be applied to a model the following key factors should be considered:

* Level of certainty – how certain does the user community need to be over the model and model outputs?
* What is the risk tolerance with regards to error?
* What are the likely sources of error? (See SCM Development Guidance for further details on types of error)
* Will the model be used on an on-going basis or for a standalone piece of analysis?
* Is the model a generic model that may be used for multiple decisions in the future?
* Who will operate the model? (Will it be the model developer or a different individual?)

### The agreed test procedures should be documented in the QA Plan and revised as required during the model development lifecycle.

## Incremental testing

### Incremental testing is the process of performing additional testing ‘on-top’ of what has already been tested. For example, if Verification was performed on version 046 of a model, and then further changes were made to the model and incorporated in version 050, incremental Verification testing may be required. ‘Incremental’ testing would only test the new changes made (between version 046 and 050) and would place reliance on the previously performed testing of model version 046.

### Incremental testing may also be applicable to models that have been tested but subsequently been adapted or repurposed to support a different decision.

### Performing incremental testing generally requires the use of software to identify the changes between model versions. Where software is unavailable it is hard to confirm with certainty exactly which changes have been made to the model and, in turn, this makes it difficult to rely on prior testing. The risk that changes made to areas of the model already tested, and not subject to incremental testing, needs to be carefully managed.

### When performing incremental testing it is important to not only test the change made, but also to check that the change does not impact on any other areas of the model already tested and introduce new issues.

### Where incremental testing is performed, Version Control is critical, as is documenting clearly what was tested in each version of the model. See section 3.4 for further details on QA Documentation.

### Depending on the nature of the changes, incremental testing may not be appropriate and it may be necessary to perform the test procedure in full. For example, if there have been substantial structural changes between model versions.

# Testing Process

The model testing process will vary depending on the test procedure(s) selected. However, Figure 2 below depicts indicative steps that are common to most model test procedures:

**Figure 2: Indicative Testing Process**

### 

### Steps 1 to 5 in Figure 2 represent one ‘Review Cycle’.

## Step 1 (see Figure 2) – The Draft model and associated documentation should be sent to the quality assurer to perform the test procedure

### The model and the associated documentation (e.g. model Specification) should be provided to the quality assurer.

### In addition, the test procedure required to be performed (as documented in the QA Plan) should be communicated to the quality assurer along with expected timescales for completion.

### The model and associated documentation should only be provided to the quality assurer once the model/documentation is ready to be tested. This is normally (although not always in the case of interim testing) towards the end of the development lifecycle, during the Test phase (see Figure 1).

### The effectiveness of testing will be affected by the absence of real data within the model as, for example, the model may only behave as intended between certain ranges/thresholds. As such, where possible the model should not be submitted for testing until it is populated with real data. For Validation (see Section 7) in particular, the model will most likely need to be populated with real data.

### The model/documentation provided by the model developer should be saved by the quality assurer and the version number of the model/documentation provided noted. This information will be included in the testing documentation (see Section 3.4 and 3.6 for further details on Documentation and File Management and Version Control).

### Once the model/documentation has been shared with the quality assurer, no further changes should be made to that version of the model/documentation by the model developer. Any changes required to the model/documentation as a result of the testing process should be made in an updated version number of the model/documentation. Changes should be kept to the minimum necessary to address the queries raised by the quality assurer.

### For example, if version 046 is shared with the quality assurer for testing, no further changes should ever be made to version 046 of the model. Any subsequent changes to the model should be made in higher versions of the model (e.g. version 047 of the model). This is to help maintain an effective audit trail.

## Step 2 (see Figure 2) – Testing performed by the quality assurer in line with QA Plan/agreed test procedures

### The quality assurer should save two copies of the model/documentation provided by the model developer for testing (see Section 3.6 for further details on File Management and Version Control); one copy for reference purposes and one copy for testing purposes. When testing the model, the quality assurer may want to change aspects of the model (e.g. flexing inputs). By saving a reference version, the quality assurer can always return to an original copy if needed.

### Testing should be performed in line with the QA Plan/agreed test procedures. This will involve testing the model/documentation against the relevant Basis for Comparison to which the testing is being conducted. See section 3.3 for further details.

### The exact procedures performed in this step will vary depending on what test procedure is being performed.

## Step 3 (see Figure 2) – Quality assurer documents queries resulting from testing and shares them with model developer

### Through the process of testing, the quality assurer will likely have a number of queries (e.g. comments, findings, observations, clarifications). All of these are likely to require some form of response by the model developer and some are likely to require changes to the model/documentation. The quality assurer should document these queries in a Query Log that can be shared with the model developer (see Section 3.5 for further details of Query Logs).

### The Query Log should be shared with the model developer so that they can respond to queries and update the model/documentation as appropriate.

## Step 4 (see Figure 2) – Queries are resolved by the model developer in an updated version of the model/documentation

### The model developer should resolve the queries raised by the quality assurer and, where necessary, make changes to the model/documentation. Any changes to the model/documentation should be made in a new version of the model/documentation (e.g. increase the version number) and updated in the model’s Version Control Log to maintain an audit trail.

### In resolving the queries, the model developer may require clarification from the quality assurer. As such, discussions may be required between the model developer and quality assurer so that the model developer clearly understands the queries raised. The drafting, by the quality assurer, of clear unambiguous queries can help to reduce the number of clarifications and, in-turn, help to expedite the testing process.

### The model developer should review each query raised and document, within the Query Log, an explanation of what has been done to address it or why no action has been taken. This is an important step; it formalises the model developer’s responses and provides a mechanism by which the quality assurer can check responses and changes in the model/documentation and, in turn, close the queries. The responses to the queries should be saved in an updated version of the Query Log file. Once all queries have been closed the quality assurer should be able to sign off the test procedure as complete.

### Changes made by the model developer to the model/documentation should be minimised and only made in relation to queries raised by the quality assurer. It should not be seen as an opportunity to ‘tweak’ other areas of the model as this would likely require additional incremental testing. It is important to limit, as far as possible, structural changes such as adding and, in particular, deleting columns or rows as this can substantially increase the time required by the quality assurer to check the changes and close the queries.

## Step 5 (see Figure 2) – The amended model and associated documentation are checked by quality assurer

### The model developer should provide the quality assurer with an updated version of the model/documentation and an updated version of the Query Log setting out responses to each query raised. The quality assurer should then check through the responses to the queries and review the changes in the model/documentation. The quality assurer should check the queries have been appropriately resolved and no new issues have been introduced as a result of changes. If satisfied, the quality assurer should mark the queries as closed.

### If the quality assurer is not satisfied with the response/changes, further iterations (review cycles) of the testing process may be required. This will require repeating the steps documented above.

### The status of each query should be updated in the Query Log and saved in a new version of the Query Log file in order to maintain an appropriate audit trail.

### The end goal of this process is to arrive at a position where all queries have been closed. In the event that some queries are awaiting closure or are not going to be closed this should be clearly documented in the Test Memo prepared by the quality assurer.

## Step 6 (see Figure 2) – Test procedure and outcomes documented in Test Memo

### In order to facilitate model sign off, the Test procedure followed should be clearly documented in a Test Memo. This will involve setting out what testing has been performed on the model/documentation, along with the findings from the tests and clearly documenting any unclosed queries or other outstanding issues. See 3.4 for further details.

# Verification

## Introduction

### This chapter covers Verification and includes:

* Overview of Verification including the underlying test procedures
* For each test procedure:
* Objectives of the test
* Basis of Comparison
* Key Information
* Key steps to perform the test
* Documentation

## Overview of Verification

### Verification is the process of checking the model for errors and alignment with its stated objectives, as set out in model documentation (e.g. the model Specification and Design). Whilst it will not provide a guarantee that all errors will be identified, it can provide a reasonably high degree of comfort in a model’s logical integrity and should be considered as part of QA planning. It involves the independent application of a broad range of analytical techniques. The specific underlying test procedures performed within Verification may vary depending on the model being tested and may include the following test procedures:

* Review of all unique formulae on a cell-by-cell basis
* Review of formula consistency within the model
* A structural review (model maps)
* Review of the overall model logic and design
* Review of inter and intra-sheet data flows
* Review of any VBA code/macros
* Review of named ranges for errors
* Review of external links for appropriateness
* A good practice critique

### Whilst the above is a list of the key test procedures it is not exhaustive and it may, on a case-by-case basis, be appropriate/necessary to perform additional test procedures.

### There is no ‘one-size fits all’ approach to Verification and it may be appropriate to perform more or less test procedures than those listed above. Performing a subset of the above procedures or only performing the above procedures on specific aspects, sections or worksheets within the model (e.g. not on the full model) would constitute ‘Partial Verification’. When performing any of the Verification procedures, it is important that the coverage is agreed between the model development team (including the Model SRO) and the quality assurer prior to testing commencing.

### Verification is recommended for sophisticated and/or high-risk models and may also be appropriate for lower risk models. For a higher degree of comfort over a model’s logical integrity it may also be appropriate to consider a Recomputation/Parallel Build. This is covered in Chapter 6.

## Review of all unique formulae on a cell-by-cell basis

**Objective of Test**

### This is the process of confirming that:

#### all unique formula in the model are operating correctly and that their application across a row, down a column or within a block is appropriate; and

#### that the model and model Specification are in alignment.

### This should include review of in-built error checks/the error check network within the model.

**Basis of Comparison**

### The Basis of Comparison for this test is twofold; comparing the formula within the model back to what has been documented in the model Specification and comparing it to the quality assurer’s knowledge of the subject area (e.g. does the formula make sense?).

**Key Information**

### A unique formula is one that is not the same as the formula in an adjacent cell.

### Depending on the number of unique formulas, their complexity and the ability to identify them (see below) this can be a time intensive exercise and the effort required to perform this test procedure should not be underestimated.

### The performance of this test procedure is best supported by spreadsheet auditing software that (as a minimum) generates a list of all unique formula in a model and their location. In the absence of software, the quality assurer would need to review every single formula, and this could increase the time taken significantly. The approach set out herein assumes software is available.

### This test should be performed in conjunction with A structural review (model maps) (see Section 5.5).

**Key steps:**

#### Using spreadsheet auditing software (e.g. available from third party providers), obtain a list of all unique formula in the model

#### For each unique formula in the list, navigate to its actual location within the model and:

#### Compare the formula to what is documented in the model Specification

#### Compare the formula to the model labelling (i.e. check the formula does what the model label suggests it should do). For example, if the model label of a row is ‘Total Staff Costs’ check the formula is calculating Total Staff Costs.

#### Check the formula links to the correct cells within the model (i.e. there are no mislinkages). For example, if a formula is “=AA12+AA13”, check that cells “AA12” and “AA13” are the correct cells the formula should be linking to.

#### Consider if the logic of the formula makes sense. This is a subjective test that involves checking the formula/formula logic against the quality assurer’s own understanding based on their knowledge and experience. For example, the formula in the model and the Specification may be “Total Indexed Costs = Total Costs x Inflation Rate”. In this example, the quality assurer may question whether the formula in both the model and the model Specification is correct or should actually be “Total Indexed Costs = Total Costs x (1 + Inflation Rate)”. Note, the quality assurer should not provide a suggested solution to the model developer (see point 3 below) and only raise a query.

#### Check that the formula’s application across a row, down a column or within a block is appropriate (this can be aided by the review of model maps – see Section 5.5).

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of formula consistency within the model

**Objective of Test**

### This is the process of confirming that all formula within the model are consistently applied. This includes:

#### Formula in calculation blocks are consistent (e.g. calculation blocks include formula that have been copied from the top left cell across to the right and down to the bottom consistently).

#### Formula between calculation blocks are consistent (e.g. where a calculation is repeated in a different area of the model a consistent methodology is applied).

**Basis of Comparison**

### Consistency within calculation blocks should be made with reference to the top left cell of each block.

### Consistency between calculation blocks should be made with reference to other repeat calculation blocks within the model.

**Key Information**

### Modelling good practice states that calculation blocks should use a consistent formula that works from left to right and top to bottom across the entire block.

### Confirming that formulae within calculation blocks are consistent is only effective if the unique formula has been confirmed as correct (see Section 5.3).

### Confirming the consistency of formulae within calculation blocks is best achieved by running and reviewing model maps (see Section 5.5).

**Key steps:**

#### When reviewing a unique formula (per section 5.3), confirm consistency between different calculation blocks performing the same calculations. For example, if one calculation block calculates ‘Total Costs’ for Division A, the same formula for ‘Total Costs’ for Division B may be required. Note, this will require an understanding of the model Specification as there **may** be reasons for differences in different calculation blocks (in this example Division A and B).

#### As part of model map reviews (see Section 5.5) confirm that there is consistency within calculation blocks.

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## A structural review (model maps)

**Objective of Test**

### This is the process of looking for formulaic inconsistences, hardcoded constants in calculation blocks, misalignments, erroneous formula etc.

**Basis of Comparison**

### The basis of comparison in this test is consistency. This test requires subjective judgement and relies on the knowledge and experience of the quality assurer.

**Key Information**

### This test requires the use of model auditing software to run model maps. Model maps provide a visual representation of the location of all labels, numerical inputs and formulas within a model. They can be used to, for example, help identify:

### Where formulae have/haven’t been copied down/across calculation blocks consistently;

### The presence of hardcoded values within areas of the model that are expected to contain formulas;

### The presence of calculations within areas of the model that are expected to contain inputs;

### The alignment/misalignment of input/calculation blocks within and across different worksheets within the model.

**Key steps:**

#### Use model auditing software to run model maps

#### For each sheet, review the maps for areas of inconsistency, including:

#### Formula that have not been copied down/across calulcation blocks consistently

#### Erroneous hardcoded values (e.g. ‘inputs’ within calculation areas)

#### Any other annomiles identified (e.g. misalignment of blocks, calculations within input areas, inputs/calculations in areas outside of the apparent used range)

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of the overall model logic and design

**Objective of Test**

### This is the process of identifying modelling issues from a logic and design perspective, including inefficiency or inflexibility. It is focussed at a level above that of individual formulas within the model.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance), the quality assurer’s knowledge and experience and the model’s documentation (e.g. model objectives, model Design).

**Key Information**

### This test looks at potential issues such as:

#### Misalignment with model Design (per model documentation)

#### Model construction/layout

#### Data points/flows

#### Calculation speed

#### Ease of use

#### Ease of update

**Key steps:**

#### Review the model’s overarching logic against that set out in the model Design

#### The quality assurer should familiarise themselves with both the model and its required objectives to assess, based on their knowledge and experience, whether they have been achieved in an efficient and effective manner.

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of inter and intra-sheet data flows

**Objective of Test**

### This is the process of confirming that the data flows are logical and efficient.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance) and the quality assurer’s knowledge and experience.

**Key Information**

### An SCM should ‘read like a book’. The logic and calculations within a worksheet should generally flow from left to right and top to bottom and within a workbook from front to back.

### Counterflows, where logic feeds upwards or from right to left within a worksheet should be avoided. Worksheets should be organised so that data flows from inputs at the front of the workbook through calculations to outputs at the back of the workbook. This helps usability and aids detection of errors in calculation logic. Acceptable deviations include summary sections within worksheets and summary output sheets at the front of the workbook, which can aid understanding and usability.

### The process of reviewing intra sheet data flows can be aided by software (some model auditing software can produce data flow maps that show the number of connections between different worksheets).

**Key steps:**

#### The quality assurer should look through the model (and data flow maps if available) and challenge both inter and intra sheet data flows in line with modelling good practice.

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of any VBA code/macros

**Objective of Test**

### This is the process of confirming that any VBA (macros) within the model work correctly and is both efficient and robust.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance), the quality assurer’s knowledge and experience and the model Specification.

**Key Information**

### VBA typically presents an elevated risk of error compared to native Excel and, as set out in the SCM Technical Build Guidance, should be avoided for performing calculations that impact model outputs.

### Where included, the review of VBA should be performed by suitably qualified and experienced specialists.

**Key steps:**

For each object, collection and module within the model’s VBA Project:

1. Create a copy of the VBA code (this may be performed via some model auditing software) against which any queries can be raised
2. Map the code logic flow (e.g. between VBA and Excel and within and between different subroutines) and check it for alignment with the model Specification.
3. Undertake a sense check of the overarching VBA logic. This is a subjective test that involves checking the logic against the quality assurer’s own understanding based on their knowledge and experience.
4. Review the code on a line by line basis checking for errors in logic/application, inefficiency, robustness and deviation from good practice.
5. Run the code under a range of conditions to confirm that it runs as intended. These conditions may include (where relevant), but are not limited to:
   1. Sheets protected/unprotected
   2. Workbook protected/unprotected
   3. Worksheets hidden
   4. Worksheets grouped
   5. Worksheets with filters on
   6. Alternative data sets

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of named ranges for errors

**Objective of Test**

### This is the process of confirming there are no errors contained within Named Ranges.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance), the quality assurer’s knowledge and experience and, with reference to the unique formula list (see Section 5.3), how named ranges are used within formulas.

**Key Information**

### Errors in Named Ranges can erroneously affect modelling outputs.

**Key steps:**

1. Create a list of all Named Ranges within the model via Excel’s Name Manager (this may be performed via some model auditing software) against which any queries can be raised
2. Check that each Named Range:
3. Covers the correct area
4. Is not erroneously linked to an external file
5. Does not include #Ref! errors
6. Is not erroneously duplicated
7. (Where appropriate) is included in model documentation
8. Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of external links for appropriateness

**Objective of Test**

### This is the process of confirming that all external links in the model are appropriate.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance), the quality assurer’s knowledge and experience and the model Specification.

**Key Information**

### It is generally advisable to avoid the use of external links. The SCM Technical Build Guidance provides details of good practice principles where external links are deemed as necessary.

**Key steps:**

1. Where appropriate, challenge the model developer on the reasons for including external links and consider if there are other, more appropriate ways to achieve the same results.
2. Create a list of all external links using Excel’s ‘Data/Edit Links’ menu (this may be performed via some model auditing software) against which any queries can be raised
3. For each external link within the model, and with reference to where it is being used within the model, check that:
4. It is in line with guidance set out in the SCM Technical Build Guidance (e.g. Called-up, Delineated, Range Named and Sign Posted)
5. It is appropriate
6. It is correctly linked
7. The link updates correctly upon opening/closing the source (feeder file) and destination file (the model being tested)
8. Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## A good practice critique

**Objective of Test**

### This is the process of confirming that the model is constructed in line with modelling good practice.

**Basis of Comparison**

### The Basis of Comparison for this test is to modelling good practice (see SCM Technical Build Guidance) and the quality assurer’s knowledge and experience.

### An SCM Good Practice Build Toolkit has been produced by Cabinet Office to support the performance of a good practice critique.

**Key Information**

### Good practice points, against which the model should be reviewed, range from avoiding the use of complex formulae to consistency of labelling and model layout. Models that fall short of good practice can be more difficult to understand and use and can present a significantly greater risk of error. This is an important review to perform when seeking to upskill and develop modelling capability within contracting authorities.

**Key steps:**

1. The model should be reviewed in a structured manner against the breadth of good practice points (see SCM Technical Build Guidance). The SCM Good Practice Build Toolkit can be used to support this test and should be consulted for further details.
2. Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Documentation

### In line with the principles of section 3.4, a Test Memo should be produced that documents what testing has been performed as part of Verification and the outcomes of those tests.

### Key information to include within a Verification Test Memo includes:

* A list of all of the tests performed (e.g. all tests in Sections 5.3 to 5.11) and the extent to which they were performed
* Details of who performed the tests, when they were performed and on what version of the model they were performed
* The key outcomes of each test

### The Test Memo should describe each test in enough detail to enable reperformance, if necessary, by an independent person.

### Key supporting documentation to the Test Memo, which should be retained on file may include:

* Query Log
* List of all Unique Formula – with comments confirming each formula has been reviewed
* List of all VBA code – with comments confirming each line of code has been reviewed
* List of all Named Ranges – with comments confirming each Named Range has been reviewed
* List of all external links – with comments confirming each external link has been reviewed
* Copies of model maps run and reviewed
* Copies of the original model and all updated model versions as a result of any additional review cycles
* Copies of model documentation provided to support the testing, such as the model Specification (inc. Design), User Guide and Developer or Technical Guide, Book of Assumptions / Data Log, and all updated versions as a result of any additional review cycles
* Copy of the final model and model documentation signed off

### More than one quality assurer may perform testing on the Model. In these cases, it may be appropriate to have a separate Test Memo, prepared by each quality assurer, to document the testing that they have each performed.

# Recomputation/Parallel Build

## Introduction

### This is a rigorous test procedure that uses recomputation to test the mathematical operation of a model. It should be considered as part of QA planning, especially given the time taken to undertake a parallel build.

### It involves the independent development of a parallel model, to the same specification, followed by a comparison of the results produced by each model.

### In some situations, it may be preferable to Verification as, for example, it can help to mitigate risks of omission (e.g. the model does not include aspects that were set out in the model Specification) and interpretation and can provide independence of mind.

### For models that require very high levels of comfort it may be also considered in addition to Verification.

### This process may require multiple iterations to complete and to understand and resolve any differences that may arise between the original model and the parallel model.

### The SCM Development Guidance provides further information on parallel build, including the circumstances under which it may be considered appropriate.

## Basis of Comparison

### Parallel model outputs are compared to the original model outputs

## Key steps

1. The model Specification is provided to an individual who is independent of the original model (they should have had no sight of the model being tested).
2. A parallel model is built in line with the model Specification.
3. The same inputs used in original model are loaded into the parallel model.
4. The outputs of the parallel model are compared with the outputs of the original model.
5. Where differences are identified they are investigated. During this process, the person who built the parallel model should remain independent from the original model and the person who built the original model should remain independent from the parallel model. This process will usually require another person to help resolve differences.
6. As differences are resolved they should be updated in the original model and/or model Specification as required. The end goal is to arrive at a position where the original model accurately reflects the model Specification.

## Documentation

### In line with the principles of section 3.4, a Test Memo should be produced that documents what testing has been performed as part of the Recomputation/Parallel Build and the outcomes of those tests.

### Key information to include within a Recomputation/Parallel Build Test Memo includes:

* The approach taken to testing and details of all of the tests performed
* Details of named individuals who performed the recomputation
* The key outcomes of each test

### Key supporting documentation to the Test Memo, which should be retained on file may include:

* Query Log
* Copies of the original model and model Specification and updated versions as a result of any additional review cycles
* Copies of the parallel model and updated versions as a result of any additional review cycles
* Copies of the comparison results for each review cycle
* Copy of the final model and model Specification signed off

# Validation

## Introduction

### This chapter covers Validation and includes:

* Overview of Validation including the underlying test procedures
* For each test procedure:
* Objectives of the test
* Basis of Comparison
* Key Information
* Key steps to perform the test
* Documentation

## Overview of Validation

### Validation is the process of checking the extent to which the model is reflective of the real-world situation and is appropriate to support its intended purpose, as set out in model documentation (e.g. the model Specification and Design). It is a test performed on the data and assumptions that feed into the model, the way that they are manipulated, and the outputs produced. It also includes a Commercial Review and Analytical Review as well as a more granular review of model input data and assumptions and model methodology.

### The specific underlying test procedures performed within Validation may vary depending on the model being tested and may include the following test procedures:

* Review of modelling methodology
* Review of data and assumptions
* Analytical Review
* Commercial Review

### Whilst the above is a list of the key test procedures it is not exhaustive and it may, on a case-by-case basis, be appropriate/necessary to perform additional test procedures.

### There is no ‘one-size fits all’ approach to Validation and it may be appropriate to perform more or less test procedures than those listed above. Performing a subset of the above procedures or only performing the above procedures on specific aspects, sections or worksheets within the model (e.g. only looking at a subset of data) would constitute ‘Partial Validation’. When performing any of the Validation procedures, it is important that the coverage is agreed between the model development team (including the Model SRO) and the quality assurer prior to testing commencing.

## Review of modelling methodology

**Objective of Test**

### This is the process of checking the extent to which the model is reflective of the real-world situation it is designed to represent and that is appropriate to support its intended purpose.

### It is an assessment of the model’s overall methodology against its intended purpose.

**Basis of Comparison**

### The Basis of Comparison for this test is the quality assurer’s knowledge and experience of the subject area.

**Key Information**

### This test looks at the overarching modelling methodology and seeks to confirm it is appropriate.

### This test should also consider the approach taken to modelling risk and uncertainty and whether this is appropriate.

**Key steps:**

#### The quality assurer should review the associated model documentation and form a view on the model objectives.

#### The quality assurer should review the overarching modelling methodology and confirm that this is in alignment with the model objectives.

#### The quality assurer should review and confirm that the model that has been built does meet the model objectives.

#### This may include checking and confirming that different options/scenarios included in the model are treated in an appropriate way, which will allow them to be compared.

#### The quality assurer should consider the approach taken to modelling risk and uncertainty and whether this is appropriate.

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Review of data and assumptions

**Objective of Test**

### This is the process of checking the extent to which the data and assumptions used within the model are appropriate. This will include an assessment of the maturity and appropriate use of data.

### This will include an assessment of the completeness of the Book of Assumptions / Data Log.

**Basis of Comparison**

### The Basis of Comparison for this test is the quality assurer’s knowledge and experience of the subject area, the Book of Assumptions / Data Log and the model Specification.

**Key Information**

### This test involves looking at the underlying data within a model and confirming its appropriateness.

### Depending on data volumes, a statistically significant data sample may be employed over full sampling.

**Key steps:**

#### The extent to which data sampling will be used in this review should be agreed and documented.

#### The quality assurer should review the associated model documentation and form a view on what the model is designed to achieve.

#### For all sampled data, the quality assurer should review the underlying data used within the model and assess the maturity and appropriateness of the data used.

#### For all sampled data, the quality assurer should review assumptions around risk and uncertainty including dependencies.

#### The quality assurer should review and assess the completeness of the Book of Assumptions / Data Log.

#### Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Analytical Review

**Objective of Test**

### This is the process of confirming the model operates in line with the intended logic.

**Basis of Comparison**

### This technique requires an initial test plan covering the input values (or value ranges) and the ‘expected’ outputs (in absolute or range and behaviour terms).

**Key Information**

### This technique is input-output focussed and examines the outputs generated by the model under different input conditions.

### It is used to inform an understanding of the model’s analytical behaviour, often without reference to the underlying calculations themselves.

### Analytical Review often makes use of graphing and charting and can be aided by the use of software.

### It is complementary to other, more granular, QA and testing approaches.

### This test procedure will require changing the model (e.g. changing inputs) and therefore should be done in a copy of the original model.

**Key steps:**

#### The quality assurer should document an initial test plan covering the input values (or value ranges) intended to be tested along with the ‘expected’ outputs (in absolute or range and behaviour terms). Note that the scope of the test plan may require input from key stakeholders, such as the model developer.

#### The quality assurer should run the inputs set out in the initial test plan through the model and confirm whether the outputs are as expected.

#### Where outputs are not as expected the quality assurer should investigate and determine the reason for this.

1. Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Commercial Review

**Objective of Test**

### This is the process of confirming the appropriateness of the model’s overarching methodology in the context of the service, project or programme being costed.

### This is the process of confirming the model’s input and output values lie within expected ranges.

### A Commercial Review aims to provide a degree of comfort at an order of magnitude level and should not be relied upon to identify model errors directly.

**Basis of Comparison**

### The Basis of Comparison for this test is the quality assurer’s knowledge and experience of the subject area.

**Key Information**

### This test requires the quality assurer to be a commercial specialist with subject matter expertise of the area being modelled.

### It is complementary to other, more granular, QA and testing approaches.

### Analytical techniques, such as regression and comparative analysis, and the construction of parallel ‘10 Line Models’ for components of the service, project or programme could be employed in support of the Commercial Review.

### This test procedure may require changing the model (e.g. changing inputs) and therefore should be done in a copy of the original model.

**Key steps:**

1. The quality assurer should review the inputs of the model and sense check them against their expectations, knowledge and understanding of the area being modelled.
2. The quality assurer should review the outputs of the model and sense check them against their expectations based on the model inputs and their knowledge and understanding of the area being modelled.
3. The quality assurer should review the appropriateness of the model’s overarching methodology in the context of the service, project or programme being costed.
4. Any queries raised should be documented in a clear and concise manner within the Query Log. In order to maintain independence and reduce the risk of self-review, the quality assurer should not provide suggested changes to model developer (see Section 3.1 for further details).

## Documentation

### In line with the principles of section 3.4, a Test Memo should be produced that documents what testing has been performed as part of Validation and the outcomes of those tests.

### Key information to include within a Validation Test Memo includes:

* A list of all of the tests performed (e.g. listing out the test procedures from sections 7.3 to 7.6 that were performed and the extent to which they were performed)
* Details of who performed the tests, when they were performed and on what version of the model, they were performed
* The key outcomes of each test

### The Test Memo should describe each test in enough detail to enable reperformance, if necessary, by an independent person.

### Key supporting documentation to the Test Memo, which should be retained on file may include:

* Query Log
* Copies of the original model and updated model versions as a result of any additional review cycles
* Copies of model versions used to perform Analytical or Commercial Review
* Details and results of all tests performed
* Copies of model documentation provided to support the testing, such as the model Specification (inc. Design), User Guide and Developer or Technical Guide, Book of Assumptions / Data Log, and all updated versions as a result of any additional review cycles
* Copy of the final model and model documentation signed off

### More than one quality assurer may perform testing on the Model. In these cases, it may be appropriate to have a separate Test Memo, prepared by each quality assurer, to document the testing that they have each performed.

# Other Testing

## Introduction

### This section covers the following additional test procedures that a model should typically be subject to:

* Developer Testing
* User Testing
* System Testing

### In general, the requirement for User Testing and System Testing will depend on how the model will be used and who will operate it.

## Developer Testing

### Having the model tested by the developer prior to formal QA and testing is an important activity that both helps to reduce the risk of error and streamline the review cycle. Prior to release for testing the developer should undertake and document a self-review of all aspects of the model and any associated documentation. This should include, although may not be limited to, checking that:

* The design logic is correct and aligns with the model Specification;
* The model is constructed in line with good practice (see SCM Technical Build Guidance);
* The in-built error checks and the error check network operate correctly; and
* The model can run all scenarios/ sensitivities and functions appropriately.

A baseline set of developer tests are included in the SCM Development Checklist. Depending on the complexity of the model, these tests should be tailored accordingly.

## User Testing

### This is the process of having a model tested by the end user(s) to confirm that it meets their requirements and that, following the application of protection, it operates as intended and its use has not been constrained. This may, for example, include checking that:

* The model produces the required outputs in the required format;
* Inputs are arranged as required and can be entered into the model;
* The model contains the required switches and they are operational;
* The overall usability and ease of navigation meets requirements; and
* The User Guide is sufficiently detailed and is readily understood.

### User testing may not be relevant if the model user is also the model developer (e.g. if the model is not being ‘handed over’).

## System Testing

This is the process of checking that the model operates in the operating environment it is intended to. System testing includes checking that, for example, the model:

* Operates as intended when loaded with real data;
* Is able to run as intended (including any required add-ins or any associated VBA code) in the operating environment; and
* Is able to connect and interface with any other files or software platforms that may be required.

# Appendix 1 – QA Report Template

**Document Control**

**Document Name:**

**Document Version:**

**Prepared by: [Insert Name & Title] Date: [Insert Date]**

**Reviewed by: [Insert Name & Title] Date: [Insert Date]**

**Approved by: [Insert Name – [Model SRO]] Date: [Insert Date]**

**Document Edit History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Comment** | **Updated By** | **Approved by** |
|  |  |  |  |  |

**QA Report for [Insert Model Name Here]**

|  |
| --- |
| **Project Background** |
| [Insert Project Background]  [e.g. In line with the proposed changes to XYZ Legislation by XYZ the XYZ Project has commenced to look at the redesign of the XXX service offering.  The new service offering is expected to go live in 202X and it is anticipated will cost in the region of £Xm per annum.] |

|  |
| --- |
| **Model Background** |
| [Insert Model Background]  [e.g. As part of XYZ Project and to support the DMA and Business Case Process an SCM has been prepared.  The model is required to be completed by dd/mm/yy and shows the expected cost of the service under x scenarios.  A new model was built to meet the requirements] |

|  |
| --- |
| **Model Configuration** |
| [Insert the File Name and Model Version]  [Insert the Model Data Configuration [e.g. Dummy Data / Draft Data / Final Data]] |

|  |
| --- |
| **IMA** |
| [Insert IMA Score]  [Insert details of QA and Testing measures suggested by the IMA] |

**Key Stakeholders**

* Key stakeholders and decision makers for the SCM are:
* [Insert Name – Insert Role]
* [Insert Name – Insert Role]
* [Insert Name – Insert Role]

[Insert as many names and roles as required]

**Model SRO**

* The Model SRO taking overall responsibility for a model and its use, including the QA and testing process is [Insert Name]

**Development Team**

The Development Team was comprised of:

* [Insert Name] – Model Developer
* [Insert Name] – Model Architect
* [Insert Name] – Quality Assurer
* [Insert Name] – Quality Assurer

[Insert additional/tailor the above roles as required considering the model/project]

**Model Development Process**

[Include details of QA and governance activities that were performed over the Model Development Lifecyle]

[e.g.

Did the development of the SCM follow a structured process that started with detailed up-front planning?

Was the model scoped, specified and designed and were the outputs of these activities documented?

Were the documents produced during the model planning and design phases signed off and approved by the Model SRO prior to the model build phase commencing?

Was the model built in line with good practice guidance?

Did the model developer undertake self-testing of the model prior to sharing the model and its associated documentation for formal QA and testing?

Have appropriate measures (e.g. Model Operator training, version and change control and planned periodic QA and testing) been put in place to help ensure that the model remains fit for purpose over its life ahead of implementation?]

**Modelling risks and mitigations**

[Insert details of specific risks associated with this modelling project and how they were mitigated. (e.g. timeline risks, data risks, complex areas in the modelling task etc.)]

**Test Procedures Performed**

The following table sets out the test procedures performed, along with the associated risk(s), the version of the model that the test procedures were performed on and the associated evidence:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk(s)** | **Test Procedure** | **Model Version Tested** | **Evidence** |
| 1 | [e.g. formulaic or arithmetic errors are included in the model] | [e.g. Review of unique formula on a cell-by-cell basis was performed covering all unique formula in the model. This test:   * checked unique formula in the model are operating correctly and in line with the model Specification * included a review of in-built error checks/error check network within the model | [e.g. Filename: Model XYZ 0048.xlsx  Version: 048  Date: dd/mm/yy  File Size: xxxx KB] | [e.g. Test performed by Joe Bloggs  See Test Memo 1 for evidence] |
| 2 | [e.g. the modelling outputs are not a realistic reflection of the project] | [e.g. a Commercial Review of the outputs was performed] | [e.g. Filename: Model XYZ 0049.xlsx  Version: 049  Date: dd/mm/yy  File Size: xxxx KB] | [e.g. Test performed by Sue Bloggs  See Test Memo 2 for evidence] |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

[Insert additional rows as required considering the model specific risk and associated test procedures performed to help mitigate these risks]

**Conclusion**

[Insert Conclusion Here – this should note whether the model has any outstanding issues (e.g. unclosed queries) or is recommended for release. Any key points from underlying Test Memos should be included here. This may include statements on risk and uncertainty in model outputs and/or on the fitness for purpose of the model]

[e.g. The above test procedures have been performed on the model and there are no outstanding issues.

It is recommended that Model XYZ version 050 can be finalised and implemented for use]

# Appendix 2 – Test Memo Template

**[Insert Model Name Here] – [Insert Description of Test Memo] – Test Memo [1]**

|  |  |
| --- | --- |
| Prepared by: [Insert Name & Title Here] | Date: [Insert Date Here] |

|  |
| --- |
| **Project Background** |
| [Insert Project Background Here]  [e.g. In line with the proposed changes to XYZ Legislation by XYZ the XYZ Project has commenced to look at the redesign of the XXX service offering.  The new service offering is expected to go live in 202X and it is anticipated will cost in the region of £Xm per annum.] |

|  |
| --- |
| **Model Background** |
| [Insert Model Background Here]  [e.g. As part of XYZ Project and to support the DMA and Business Case Process an SCM has been prepared.  The model is required to be completed by dd/mm/yy and shows the expected cost of the service under x scenarios.  The model SRO is X and the model has been developed by Y  SME input was provided by A, B and C  The model was built using the Cabinet Office Sourcing Programme Model Build Template] |

[The following is an illustration of what a Test Memo may include. The following sections should be updated to reflect:

* The test procedures performed
* The findings from the test procedures
* The number of review cycles completed; and
* The conclusions from undertaking the testing.

Each test procedure will vary, and there may be a need to perform any number of review cycles depending on the findings from the testing.

The conclusions section should make clear if there are any important outstanding issues raised through the review process (e.g. unclosed queries) or if the test procedure is complete with no outstanding issues.]

**Test Procedures Performed – Review Cycle 1**

The following procedures were performed on the following version of the model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Version** | **Date** | **File Size** |
| *Filename: Model XYZ 0046.xlsx* | *046* | *[dd/mm/yy]* | *[xxxx KB]* |

1. Review of unique formula on a cell-by-cell basis
2. Review of formula consistency within the model
3. A structural review (model maps)
4. Review of the overall model logic and design

**Findings – Review Cycle 1**

All queries from the review were documented in Query Log 001.xlsx.

The model was updated in response to queries and updated in the following version of the model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Version** | **Date** | **File Size** |
| *Filename: Model XYZ 0047.xlsx* | *047* | *[dd/mm/yy]* | *[xxxx KB]* |

All query responses and changes to the model were reviewed.

14 out of 16 comments were closed, however 2 queries remained open and were sent back to the model developer to be actioned.

The outstanding queries were tested in Review Cycle 2.

**Test Procedures Performed – Review Cycle 2**

The following procedures were performed on the following version of the model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Version** | **Date** | **File Size** |
| *Filename: Model XYZ 0048.xlsx* | *048* | *[dd/mm/yy]* | *[xxxx KB]* |

1. Review of the changes made in relation to the open comments in Review Cycle 1.

**Findings – Review Cycle 2**

The Query Log was updated and the final version was saved as Query Log 002.xlsx.

All queries are closed and there are no outstanding issues.

**Conclusion**

All review queries have been closed in the following version of the model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Version** | **Date** | **File Size** |
| *Filename: Model XYZ 0048.xlsx* | *048* | *[dd/mm/yy]* | *[xxxx KB]* |

**Attachments**:

|  |  |
| --- | --- |
| **Filename** | **Description** |
| Model XYZ 0048.xlsx | Model provided for Review Cycle 1 |
| Model XYZ 0049.xlsx | Model provided for Review Cycle 2 |
| Query Log 001.xlsx | Query Log for Review Cycle 1 |
| Query Log 002.xlsx | Query Log for Review Cycle 2 |
| Unique Fromuale.xlsx | List of Unique Formula in the model for Review Cycle 1 |
| Model Maps.xlsx | Model maps for Review Cycle 1 |



**© Crown copyright 2021**

This publication is licensed under the terms of the Open Government  
Licence v3.0 except where otherwise stated. To view this licence,  
visit [nationalarchives.gov.uk/doc/open-government-licence/version/3](http://nationalarchives.gov.uk/doc/open-government-licence/version/3/)

Where we have identified any third party copyright information you  
will need to obtain permission from the copyright holders concerned.