



Government  
Commercial  
Function

# SHOULD COST MODELLING

## Development Guidance

MAY 2021

Version 1.0

# Contents

<b>1.</b>	<b>Foreword</b>	<b>3</b>
1.1	Overview	3
1.2	Contact	4
<b>2.</b>	<b>Background</b>	<b>5</b>
2.1	Guidance Context	5
2.2	Purpose & Intent	5
2.3	Application & Scope	6
2.4	Responsibility and Accountability	7
<b>3.</b>	<b>Should Cost Model Development Lifecycle</b>	<b>9</b>
3.1	Defining the Should Cost Model	9
3.2	Process Overview	10
3.3	Development Options – Reuse, Reconfigure or New Build	11
<b>4.</b>	<b>Risk Management</b>	<b>13</b>
4.1	Effectively managing risk	13
4.2	Design Risks	13
4.3	Error Risks	14
4.4	Access Risks	14
4.5	Availability Risks	15
4.6	Errors of Omission	15
<b>5.</b>	<b>Designing a Should Cost Model</b>	<b>16</b>
5.1	Overview of Model Specification inc. Design	16
5.2	Key Components of a Model Specification inc. Design	17
5.3	QA, Data and Delivery Plans	19
<b>6.</b>	<b>Should Cost Model Development</b>	<b>21</b>
6.1	Overview of Development	21
6.2	Build to Good Practice	21
6.3	Data Population	22
6.4	Integral Model Checks	22
6.5	Developer Testing	23
6.6	User Guide	23
6.7	Technical or Developer Guide	24

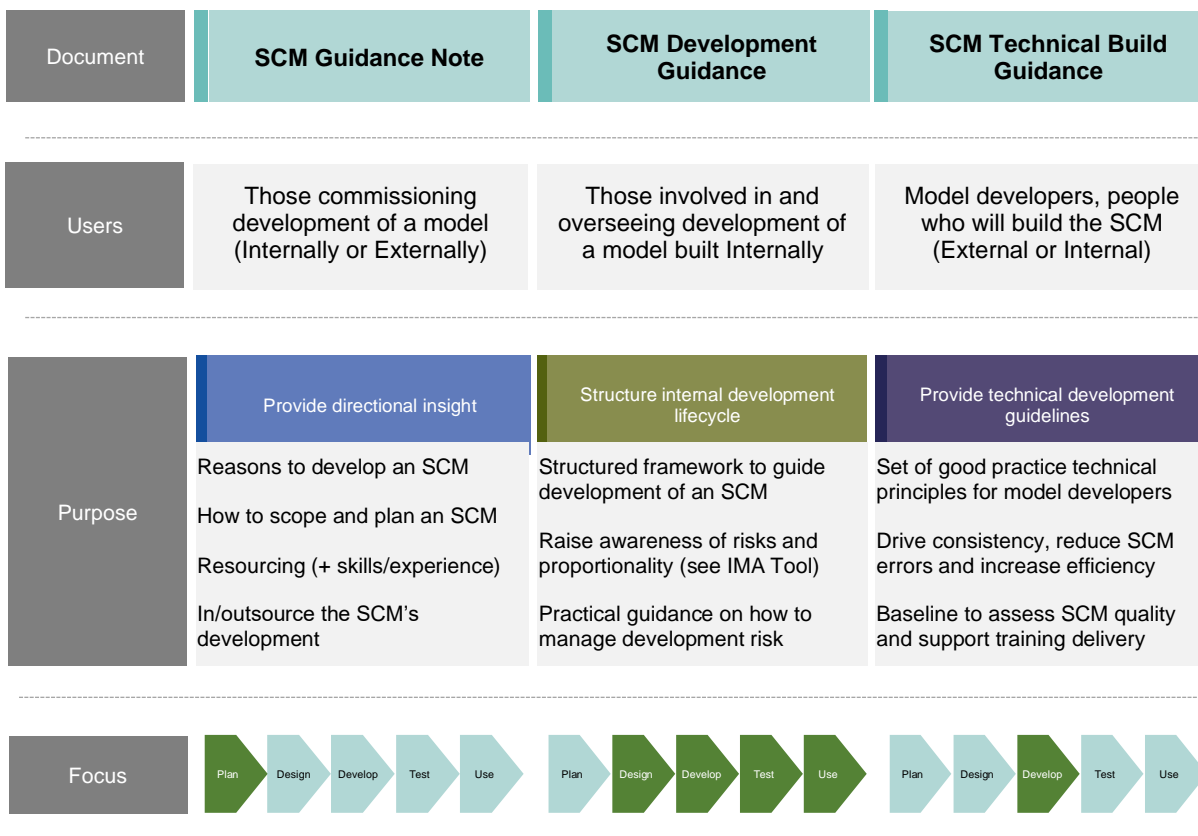
6.8	Re-use of Previously Developed models	24
6.9	Book of Assumptions / Data Log	24
6.10	Development Sign-Off	25
<b>7.</b>	<b>Quality Assuring and Testing</b>	<b>27</b>
7.1	Overview of Quality Assurance and Testing	27
7.2	Verification and Validation	27
<b>8.</b>	<b>Should Cost Model Use</b>	<b>29</b>
8.1	Governance	29
8.2	Handover	29
8.3	Tracking	29
8.4	Stakeholder Mapping	30
8.5	Revisions, Re-Use and Sign-Off	30
8.6	Periodic Testing	31
<b>9.</b>	<b>File Management</b>	<b>32</b>
9.1	Overview	32
9.2	File Management Considerations	32
<b>10.</b>	<b>Tools, Templates &amp; Checklists</b>	<b>34</b>
10.1	Overview	34
10.2	Planning and Design Templates	34
10.3	Model Build Template	35
10.4	Version Control Log	35
10.5	Configuration Control Log	35
10.6	Review Tools	35
10.7	Good Practice Checklist	36
10.8	Development Checklist	36
<b>11.</b>	<b>Appendix I: Quality Assurance and Testing</b>	<b>37</b>
<b>12.</b>	<b>Appendix II: Roles &amp; Responsibilities</b>	<b>43</b>
<b>13.</b>	<b>Appendix III: Glossary &amp; Abbreviations</b>	<b>44</b>
<b>14.</b>	<b>Appendix IV: Designing a Model</b>	<b>45</b>
<b>15.</b>	<b>Appendix VI: Cost Estimating Techniques</b>	<b>50</b>

# 1. Foreword

## 1.1 Overview

- 1.1.1 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](#) and for **public works projects or programmes** within the [Construction Playbook](#).
- 1.1.2 The Sourcing and Construction Playbooks set out when contracting authorities should produce an SCM and how SCMs fit within the procurement lifecycle. The accompanying [SCM Guidance Note](#) provides further guidance on what SCMs are, why contracting authorities should produce them and how to commission them.
- 1.1.3 This publication provides contracting authorities with guidance on how to design, develop, test and manage SCMs. As shown Figure 1, it is part of a suite of SCM guidance produced by Cabinet Office, which also includes good practice guidance for building SCMs, within the SCM Technical Build Guidance.

**Figure 1: Cabinet Office SCM Guidance Documents**



- 1.1.4 This publication is intended to provide insight into good practice to internal resources who are involved in the design, development, testing, use or management of an SCM. This extends beyond the model developer to a broad

range of stakeholders, such as model customers and those who may be involved in or responsible for:

- Managing and overseeing the development of SCMs;
- Providing data and assumptions that underpin SCMs;
- Performing Quality Assurance (QA) and testing of SCMs; and
- Running and using SCMs to inform decision making processes.

1.1.5 This publication focusses on raising awareness of the risks of developing and using SCMs and on providing a good practice framework to help manage these risks. It has been developed through collation of good practice recommendations, including those emerging from existing government guidance<sup>1</sup>, and providing more detailed, operational, commentary on how to apply them.

1.1.6 This publication builds on the [SCM Guidance Note](#). It is recommended that the SCM Guidance Note is read, and the activities within it completed, prior to reading and following the recommendations within this publication.

1.1.7 The guidance herein 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.

## 1.2 Contact

1.2.1 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).

---

<sup>1</sup> Notably the [Green Book](#), The [Aqua Book](#), and the [Review of Quality Assurance of Government Models](#).

## 2. Background

### 2.1 Guidance Context

- 2.1.1 The development of SCMs often requires considerable time and resource investment. Ensuring that appropriate personnel are developing well-designed models in a manner that is informed by good practice approaches and techniques will enable this investment to yield greater returns, by driving accuracy, efficiency and understanding within the business.
- 2.1.2 As well as being able to produce a model that provides insight, it is important to be able to identify the risks associated with SCMs and to manage them effectively, which is the focus of this publication.

### 2.2 Purpose & Intent

- 2.2.1 The purpose of this publication is to raise awareness of the risks associated with SCMs and to provide a structured framework to help manage the development of SCMs in a way that helps to mitigate these risks.
- 2.2.2 The activities discussed within this publication draw on recommendations contained in HMT's [Aqua Book](#), the [Review of Quality Assurance of Government Models](#), and the [Green Book](#). The purpose of this document is to provide a deeper understanding and more operational insight to people developing and overseeing the development of SCMs in line with these recommendations.
- 2.2.3 This document raises awareness of the benefits of adopting a robust and consistent approach to the management of SCMs and their associated risks. There are many benefits from structuring model development, including:
- **Risk Identification** – helps to ensure that risks are identified early and appropriate mitigating actions are taken;
  - **Risk Prioritisation** – helps to ensure that risks are effectively prioritised and addressed during model development and prior to model use;

---

“Ensuring that appropriate personnel are developing well-designed models in a manner that is informed by good practice approaches and techniques will enable this investment to yield greater returns”

---

- **Increased Efficiency** – helps to reduce duplicated effort and enable efficiencies through process improvement, automation and by limiting iteration and rework;
- **Greater Focus** – helps to increase model specificity and keep it aligned to the needs of the customer by following a structured development plan with regular engagement across the stakeholder population; and
- **Confidence** – helps to manage model development risk and allow for greater confidence in decisions supported by SCMs.

## 2.3 Application & Scope

- 2.3.1 The guidance within this publication is applicable to all SCMs and the files that underpin them, such as feeder models or data manipulation and analysis files.
- 2.3.2 It is primarily aimed at spreadsheet-based models, such as those developed using Microsoft Excel. However, the general principles are applicable to SCMs developed using other software, such as databases.
- 2.3.3 It should be applied in a manner that is proportional to the risks associated with a specific SCM and its use. It is part of a suite of SCM guidance produced by Cabinet Office, which includes:
- [SCM Guidance Note](#) – outlines what SCMs are, when and why contracting authorities should produce them, and key considerations around developing and/or procuring them;
  - **Initial Model Assessment Tool** – a tool to help inform the SCM development approach and help contracting authorities to set an appropriate, proportional, level of QA and testing. It is informed by the Cabinet Office [Tiering Tool](#), which should be used to provide an initial indicator of the criticality to the business of the decision that the SCM is designed to support; 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.
- 2.3.4 A number of practical Tools and Templates have also been produced by Cabinet Office to support the development of SCMs and to help reinforce good practice approaches. These, together with the guidance set out above, are aligned to different phases/ stages of the model development lifecycle (see Figure 2).

**Figure 2: Model Development Lifecycle (Including SCM Governance Process Overview and Supporting Cabinet Office Guidance, Tools and Templates)**

Project Phase / Stage	Quality Assurance					
Activities	Plan	Design	Develop	Test	Use	
Description	<ul style="list-style-type: none"> <li>Assessment of inherent model risk performed</li> <li>Assess criticality of the decision and level of sophistication of the supporting model</li> </ul>	<ul style="list-style-type: none"> <li>Requirements documented</li> <li>Selection of appropriate tool / software platform</li> <li>Plans for delivery, resourcing and quality assurance prepared</li> </ul>	<ul style="list-style-type: none"> <li>Formalised model deliverables, outputs and inputs in prototype model</li> <li>Agreed model logic / methodology</li> <li>Codified data plan</li> </ul>	<ul style="list-style-type: none"> <li>Model built following good practice guidance</li> <li>Data prepared and populated</li> <li>Developer testing performed</li> <li>Additional documentation produced</li> </ul>	<ul style="list-style-type: none"> <li>Appropriate QA and testing performed</li> <li>QA documentation prepared</li> <li>Release processes and requirements followed</li> </ul>	<ul style="list-style-type: none"> <li>Decision support sign-off</li> <li>Model maintenance</li> <li>File management</li> </ul>
Key Documentation to Produce*	<ul style="list-style-type: none"> <li>Initial Model Assessment</li> </ul>	<ul style="list-style-type: none"> <li>Model Scope</li> <li>Delivery Plan</li> <li>QA Plan</li> </ul>	<ul style="list-style-type: none"> <li>Model Specification (inc. Design)</li> <li>Data Plan</li> </ul>	<ul style="list-style-type: none"> <li>Draft Model</li> <li>Book of Assumptions / Data Log</li> <li>Updated Model Specification (inc. Design)</li> <li>[User Guide]</li> <li>[Technical Guide]</li> </ul>	<ul style="list-style-type: none"> <li>QA Report</li> <li>Supporting Test Memos</li> </ul>	<ul style="list-style-type: none"> <li>Updated Model</li> <li>Updated Book of Assumptions / Data Log</li> <li>Updated Model Specification (inc. Design)</li> <li>[Updated User Guide]</li> <li>[Updated Technical Guide]</li> </ul>
Key Sign Offs	IMA Tool	Scope / QA Plan	Specification & Design	Draft Model	QA	Final Model / Documentation
Guidance Tools and Templates	SCM Guidance Note Initial Model Assessment Tool	SCM Guidance Note Scoping Template Planning Template QA Plan Template Development Checklist	Development Guidance Technical Build Guidance Specification Template Example SCM Build Template Book of Assumptions / Data Log Template Development Checklist	Development Guidance Technical Build Guidance SCM Build Template Book of Assumptions / Data Log Template Good Practice Build Toolkit Version Control Log User Guide Example Development Checklist	Development Guidance Technical Build Guidance Testing Procedures Book of Assumptions / Data Log Template Good Practice Build Toolkit Development Checklist	Development Guidance Book of Assumptions / Data Log Template Development Checklist

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

## 2.4 Responsibility and Accountability

2.4.1 The Model Senior Responsible Owner (Model SRO) is ultimately accountable for the SCM and its use. In practical terms, empowering a Model SRO to sign-off an SCM for use may best be achieved by implementing a structured framework for the development of SCMs within contracting authorities. This publication, together with the accompanying guidance, tools and templates produced by the Sourcing Programme can inform that framework.

2.4.2 In addition to having a framework in place, it is important to ensure that SCMs are both developed by suitably qualified and experience personnel and developed in line with the framework. Inevitably this will require someone to oversee the model development process and to confirm adherence to the framework. Whilst this role could be performed by the Model SRO it may be more appropriate for the Model SRO to delegate responsibility for oversight.



- 2.4.3 This publication will help overseers to identify what they should be looking for and the accompanying SCM tools and templates produced by the Sourcing Programme (see Section 10) will help them to discharge their responsibilities.
- 2.4.4 Notably, the guidance herein should be applied in manner that is proportional to the risks associated with a specific SCM and its use. Any decisions over the extent to which it is applied should be confirmed as acceptable by the Model SRO.

## 3. Should Cost Model Development Lifecycle

### 3.1 Defining the Should Cost Model

3.1.1 A Should Cost Model provides a forecast of what a service, project or programme 'should' cost over its whole life. As summarised below, there are different types of SCMs that may also differ in design as requirements change over the procurement lifecycle. However, the term Should Cost Model or SCM is used throughout this publication to mean all of them.

3.1.2 For public works projects, SCMs forecast costs over a period that includes both the build phase and the expected design life. This includes costs of additional market factors such as risk and profit. It provides an understanding of whole life costs, including the impact of risk and uncertainty on both cost and schedule. Notably, the key factor is 'whole life cost' and not the initial purchase price.

3.1.3 For public services an SCM can be used to help evaluate different delivery model options:

- **In-house** – This (also referred to as a 'Public Sector Comparator') is the whole life cost to deliver a service in-house using internal resources and expertise. It includes the cost of acquiring and maintaining assets and the necessary capability.
- **Expected Market Cost** – This is the expected whole life cost of procuring a service from an outside supplier. It includes the cost of additional market factors such as risk and profit.
- **Mixed Economy** – A delivery model will often be a combination of insourcing and outsourcing different components of the service. In these cases, a combination of the 'In-house' and 'Expected Market Cost' options, referred to as a 'Mixed Economy' option, can be used to calculate the cost of the service.

3.1.4 SCMs should be used early in the procurement process to:

- Inform the delivery model assessment (DMA)<sup>2</sup>, which considers both cost and non-cost criteria;
- Drive a better understanding of the whole life costs and the risks and opportunities associated with different options and scenarios;
- Drive more realistic budgets by providing greater understanding of the impact of risk and uncertainty on both cost and schedule;

<sup>2</sup> See [Sourcing Playbook](#) and [Construction Playbook](#) for guidance on delivery model assessments.

- Inform the first business case (Strategic Outline Case for departments and ALBs); and
- Inform engagement with bidders and the appropriate commercial strategy, including methods to incentivise the supply chain to focus on whole life value.

3.1.5 An SCM will evolve over time as more information becomes available and different demands are placed upon it. For example, if pursuing an external procurement route, a new SCM may need to be developed or the structure of an initially developed SCM may need to be evolved to enable insight into bids from suppliers and to help protect government from low-cost bid bias (see Chapters 10 and 9 in the [Sourcing](#) and [Construction](#) Playbooks respectively). In these cases, early market engagement should be used to help ensure that the model structure will enable comparison to the bids expected from the market.

3.1.6 The [SCM Guidance Note](#) provides further details on what a Should Cost Model is, and the commercial reasons for using one.

## 3.2 Process Overview

3.2.1 The development of an SCM should follow a well-structured and clearly defined lifecycle, with stage-specific procedures and controls (see Figure 2). These can be summarised as:

- **Proportional** – The investment in planning, designing, developing and testing a model should be proportionate to the criticality of the decision;
- **Structured** – Process stages should be structured and roles should be clearly defined and separated (e.g. separating SCM build from formal QA and testing);
- **Sequential** – The development of a model should follow sequential stages, with each stage being completed before commencing the subsequent stage (e.g. completing model design before starting model development); and
- **Signed Off** – Stages should be agreed and signed off prior to progressing. This can be facilitated through the use of checklists (Section 10).

---

“An SCM will evolve over time as more information becomes available and different demands are placed upon it”

---

3.2.2 With reference to the model development lifecycle (see Figure 2), this publication covers the design, development, testing and use phases. Initial SCM planning is addressed in the [SCM Guidance Note](#), which, in summary, recommends the following activities prior to designing an SCM:

- **Performing an Initial Model Assessment** – this considers both the criticality of the decision that the model is designed to support and the inherent risk in order to inform the level of governance, QA and testing required to help manage development risk. An IMA Tool has been developed by Cabinet Office to support this assessment. It is informed by the Cabinet Office [Tiering Tool](#), which should be used to provide an initial indicator of the criticality to the business of the decision that the SCM is designed to support; and
- **Undertake model planning** – the planning phase includes developing a model Scope that, for example, covers its purpose, key features, relevant costs, modelling techniques employed and the tool/software that will be used to undertake the modelling. It should also include preparation of an initial Data Plan, setting out provisional data requirements; a Delivery Plan, setting out resource requirements, timelines, risks and mitigations; and a QA Plan, setting out QA and testing requirements (informed by the Initial Model Assessment) and how the model will be governed and controlled over its lifecycle.

3.2.3 As the procurement progresses and the demands upon the SCM change it will likely need to evolve. In some cases, this may be achieved by evolving the existing SCM and in others it may be more appropriate to develop a new SCM. In either case the model development lifecycle will repeat and should follow a similar structured approach (see Section 3.3).

### 3.3 Development Options – Reuse, Reconfigure or New Build

3.3.1 Previously developed models may be available to provide decision support. When approaching a modelling task, assessing the suitability of existing models to provide relevant insight is recommended in the first instance. There is potential to deliver efficiencies through utilising existing models and reducing development effort.

3.3.2 When undertaking a review of available models, it is important to assess compatibility by reference to:

- **Relevance** – how closely related is the current to the previous decision, by reference to the nature of the costs and composition of the service offering;
  - **Detail** – is the analysis in the previously developed model at the required level of detail to support the current decision; and
  - **Functionality** – does the available model contain the required functionality to enable the analysis required to support the current decision.
- If directly compatible models are available (i.e. models that have been used to support a similar decision) their re-use may be appropriate. Partial compatibility may be evidenced

in a model that shares some common features with the model that is set to be developed in support of a decision. It may be appropriate to reconfigure these models.

- Availability of a model does not automatically mean that it should be used in support of a decision. It may be appropriate to build a new model if available models are incompatible with the current requirements.
- Whether re-using, reconfiguring or building a new model, it is important to follow the process outlined below, being aware of the principle of proportionality. Planning, designing and testing activity may have been performed in preparing a model previously. However, this does not mean that the activity does not need to be performed when reusing or reconfiguring a model. The steps still need to be completed although they may differ in nature. For example, confirming that a previously produced QA Plan remains relevant for a model that is going to be reused rather than preparing a new QA Plan to support the development of a new model.

## 4. Risk Management

### 4.1 Effectively managing risk

- 4.1.1 One of the key reasons to structure the development of SCMs is to manage risk. Models are primarily used as decision support tools and incorrect decisions can be made or other adverse impacts may be caused if one or more of a broad range of risks materialise.
- 4.1.2 The impacts of a risk materialising are not restricted just to the financial, business or economic consequences of making an incorrect decision. Non-financial impacts can extend to the legal, compliance or reputational spheres and a broad awareness of different types of risks is required in order for them to be effectively managed.
- 4.1.3 The effective management of SCM risks is not a bolt-on activity and should be integral to ways of working. It should be embraced by a positive and supportive culture where individuals are encouraged and empowered to identify and manage risks (see [Aqua Book](#)).
- 4.1.4 Adopting good practice SCM development that includes a structured approach with proportional controls founded on a consistent risk assessment, with clear documentation, ownership and accountability, provides a mechanism to manage risks across the model development lifecycle (see Figure 2).

### **The holistic management of risks across the development lifecycle requires structured controls that span initial planning activity, through design, development, testing and use.**

There are many types of risk that can impact a model. Whilst formal QA and testing helps manage risks associated with model errors or inaccuracies, there are many other risks that exist as models are developed that require management.

### 4.2 Design Risks

- 4.2.1 Models are designed for a particular purpose. They use specific calculations on defined sets of data in order to produce discrete outputs. There are risks that models may be used for purposes other than those for which they have been designed and their outputs may not be sufficient to support the decision. The design phase (Section 4) of model development codifies the model's capabilities and limitations and provides a clear and transparent record of what analysis the model has been designed to provide, what data it will contain and how it will operate.

## 4.3 Error Risks

4.3.1 Model error is wider than user error, where a model operator corrupts the model in some way, and has a broader application. Errors can be broadly categorised as follows:

- **Data Errors** – incorrect source data being used; low maturity or quality data; inappropriate data; unit translation errors; data entry errors; etc.;
- **Calculation Errors** – including coding errors, such as pointing errors, range errors, signage errors; logic errors; overtyping errors; etc.;
- **Usage Errors** – including versioning errors; configuration errors; copy-paste errors; operation errors; etc.;
- **Interpretation Errors** – where a model's outputs are misinterpreted or the model is used to support a decision it has not been designed to support; and
- **Errors of Omission** – where model elements are missing because, for example, they were not scoped or developed or input data is incomplete.

4.3.2 Data errors can be managed through preparing a Data Plan, using and maintaining a Book of Assumptions / Data Log (Section 4) and applying appropriate Quality Assurance and testing (Section 7). Reducing the likelihood of calculation errors arising can be achieved through good practice development approaches (Section 6) and a robust approach to Quality Assurance and testing (Section 7). Usage errors can be managed through developing a model in line with good practice approaches and providing sufficient documentation and training (Section 6) as well as adopting a regimented approach to file management (Section 9). Interpretation errors can be managed through model documentation and training (Section 6) as well as effective management of the model when in use (Section 8). Omission errors can be managed through good practice approaches to model planning (see SCM Scoping in [SCM Guidance Note](#) and Section 4) together with a robust approach to Quality Assurance and testing (Section 7).

## 4.4 Access Risks

4.4.1 SCM's may contain sensitive information that should only be available to a restricted audience. As well as changes in legislation (e.g. GDPR) the nature of government procurement and the sensitivity of the data used to model scenarios, or the scenarios themselves, means that improper access and use of a model is a risk that requires managing. Broad categories of access risks include:

- **Hidden Data** – data can be contained in hidden rows, columns, worksheets, comments, VBA code and can potentially provide unintended access to data;
- **Protection Failure** – providing inadvertent access to sensitive information through ineffective protection or failures in passwords (e.g. password is not applied, too simple or ineffective, or shared with an unauthorised user);
- **Access Control** – providing access to data through inappropriate access controls (e.g. storing an SCM on a shared drive, not restricting access permissions, sending SCMs via email); and

- **Data Control** – providing sensitive data to unauthorised persons. People may be authorised to use the model but not some of the data within it (e.g. GDPR requirements).

4.4.2 Documenting a list of permitted users, identifying and agreeing an environment to store the model that limits access to authorised persons, and putting in place access and sharing protocols will help to mitigate the above risks (Section 9.2).

## 4.5 Availability Risks

4.5.1 If SCMs are not available when required they can cause delay. In certain circumstances this could result in significant operational, financial and/or reputational impact to the contracting authority. Broad categories of availability risks include:

- **Loss** - this includes file loss, file corruption and file access loss, including through loss of passwords;
- **Continuity** - this includes loss of knowledge of how to operate or maintain an SCM, including through loss of key personnel or documentation;
- **Compatibility** – this includes inability to operate the SCM owing to software changes or differences between developer and user environments; and
- **Delay** - this includes late deployment through inappropriate planning, data availability delays, development delays and/or QA and testing delays.

4.5.2 Adopting a regimented approach to file management (Section 9), producing model documentation (Section 6.6, 6.7 and 6.9), confirming the operating environment ahead of development and undertaking thorough up front planning (Section 4 and 14) and developing models in line with good practice approaches (see SCM Technical Build Guidance) will help to mitigate the above risks.

## 4.6 Errors of Omission

4.6.1 SCMs need to appropriately reflect the costs associated with the underlying decision. Their scope needs to be sufficiently broad and they need to be complete. They should reflect the scope and should not have unpopulated areas or contain calculations that have not been fully developed. Errors of omission can be divided into two categories:

- **Partial Omission** – this is where only an element of the total requirement has been included within the model. For example, not populating all input years within a model.
- **Complete Omission** – this is where a key consideration has not been included within the model. This error type is not restricted to the development of a model in totality. For example, the failure to identify a relevant cost when planning and designing the model is a complete omission, as is a failure to develop elements of the calculation logic that are contained within the model's Specification.



## 5. Designing a Should Cost Model

### 5.1 Overview of Model Specification inc. Design

- 5.1.1 During the design phase (see Figure 2), the model Scope will be converted into an analytical plan that will consider the inputs, analytical methods, and the outputs that the SCM will produce (see HMT [Aqua Book](#)). This can be achieved through the development of a model Specification (inc. Design). The core components of which are outlined below and a process guide for preparing a model Specification is contained in Appendix IV: Designing a Model to this publication.
- 5.1.2 The purpose of preparing a model Specification (inc. Design) is to allow for discussion on the model's inputs, outputs, calculation logic and overall structure. Obtaining agreement between the customer, developer and other key stakeholder populations prior to undertaking development reduces the likelihood of a model being iterated, changed or updated through the development phase of the lifecycle. It also helps to ensure that the requirements of a broad range of stakeholders are reflected and that the overarching model design is fit for purpose and effective. A model Specification is often prepared in a text document (e.g. MS Word) although some aspects, such as model input and output templates, may be developed in a spreadsheet-based application (e.g. MS Excel).
- 5.1.3 The model design phase helps to manage several risks. Risks relating to the design and interpretation of a model can be addressed through detailing key decisions in a document that covers the model's purpose, functionality, data sets, calculation logic and overall layout.
- 5.1.4 It is important to consider the model Specification (inc. Design) holistically as iteration may be required to account for dependencies. For example, if some input data sets are subject to uncertainty it may be appropriate to employ more sophisticated modelling techniques, such as sensitivity or statistical analysis, to provide appropriate insight to decision makers (see the [Aqua Book](#) for a further information on uncertainty).

### **Iterations need to be appropriately managed and changes should be reflected in model documentation.**

Model development iterations are commonplace and can be caused by a broad range of factors. What is important, is that documentation, such as the model Specification, is maintained as the model progresses through development. Ultimately, the model Specification, or key aspects of it, will need to accompany or be incorporated within the model to inform users of the model's purpose and how it functions.

## 5.2 Key Components of a Model Specification inc. Design

- 5.2.1 The model Specification (inc. Design) will build upon the model Scope prepared during the initial planning phase (See Figure 2). Whereas the focus of the model Scope is setting out **what** the model is required to do the focus of the model Specification (inc. Design) is setting out **how** this will be achieved. It provides the blueprint for developing the model and, as such, will be more granular than the model Scope.

Component	Detail
<b>Purpose</b>	The model Scope will summarise the purpose of the model, what decisions it is designed to support as well as any limitations impacting the model. It is recommended that this section of the model Scope is extracted and incorporated into the model Specification to limit potential divergence. Doing this allows for development and review of the model Specification to have a reference back to the agreed upon model Scope.
<b>Model Schematic</b>	The model Specification should include a diagram or schematic that shows how the SCM will be designed and organised (the model Design). This includes the arrangement of input data and how it will be transformed through calculations into outputs. This is a visual depiction of the model's architecture, data flows and how it will operate. It is invaluable for assessing and communicating the overall design ahead of model build. It provides guidance to developers on how the different sections of a model interact, assists model quality assurers and, ultimately, can help model operators to understand and navigate the model.
<b>Calculation Logic</b>	How the model will perform key calculations should be documented within a dedicated section of the model Specification. This will increase transparency and facilitate communication and agreement with key stakeholders ahead of development, thus reducing 'black-box' syndrome. Crucially, it will also allow quality assurers to compare the model's actual calculations against the intended calculations. It will also enable end users to readily understand the mathematical operations applied, help provide confidence in the model's operation and usefully support model handover and maintenance.
<b>Advanced Analytical Techniques</b>	For more sophisticated models, more advanced analytical techniques may be required. For example, statistical techniques such as Monte Carlo simulation, accounting approaches such as Net Present Value or the application of Economic or Financial modelling techniques. These requirements, and how they will be implemented, should be captured in the model Specification. Details of any data required to perform the analysis should be captured in the Data Plan (see below) and, where specialist input is required, this should be reflected in the resourcing section of the Delivery Plan, which should have been prepared during the model planning phase.

Component	Detail
<b>Advanced Model Functionality</b>	Models may use VBA to provide advanced functionality, such as automating manual activities to reduce operator time and/or risk. The SCM Technical Build Guidance recommends avoiding use of VBA or Macros for calculation purposes. However, if the use VBA is deemed appropriate then it should be documented in the model Specification and in sufficient detail to enable unambiguous development.
<b>Application of Cost Estimating Principles</b>	What techniques will be used to estimate costs and how they will be applied at a more granular level should be considered in producing a model Specification. There are a wide range of estimation techniques and considerations (see Appendix VI: Cost Estimating Techniques) and their selection will ultimately determine the structure of the model. Inappropriate decisions may render cost estimates unfit for purpose and it is important to engage suitably qualified and experienced personnel at the right time (see Appendix II: Roles & Responsibilities for resources).
<b>Scenarios and Options</b>	The model Specification should set out what scenarios and options the model will need to produce outputs for. If required, they need to be documented and designed. Having an understanding of the requirement, or potential requirement, will better enable business intelligence to be captured in a single model. The alternative of having multiple models with different configurations can increase the risk of error, increase the administrative burden and increase the Quality Assurance and testing workload.
<b>Outputs</b>	The model's outputs should be set out within the model Specification. Typically, this might include example or dummy outputs in a spreadsheet-based format. These could be financial tables, graphs and charts or other visualisations. The outputs in the model Specification should be reviewed by the customers of the model and agreed prior to commencing further development activity. Codifying what outputs the model will produce as part of the model Specification will reduce the likelihood of additional output functionality being required after model development and testing has completed. This will help to ensure that the overall design is efficient and help to negate the risks associated with model changes, including through the potential inability to perform adequate testing within project timescales.
<b>Input Data Requirements</b>	<p>The model Specification should document all of the data and assumptions that are required for the model to produce analysis and insight. This will be used to update the Data Plan (prepared during model planning) and, in turn, inform discussions around data availability (and any associated issues). It will also support agreement with key stakeholders on the data and assumptions that will be used to power the model before it is developed.</p> <p>The Data Plan will ultimately form the basis for the model's Book of Assumptions / Data Log, which should be maintained throughout the</p>

Component	Detail
	<p>model's life so that it represents a complete list of (and provides provenance to) all of the data and assumptions used by the model (see Section 6.9).</p> <p>Production of appropriately labelled data input templates (typically in a spreadsheet-based format) can help to reduce the risk of requirements misinterpretation and can also guide data collection by helping to ensure that data is received in the required format. Ultimately, model input templates (together with model output templates) will form the basis of the model to be developed.</p> <p>Once input data requirements have been identified it is important to ensure that data is available, at the required quality and within the required timescales.</p>
<b>Key Model Drivers</b>	<p>The model Specification should document the key drivers of the model. Drivers are the parts of the model that influence the required levels of a particular service, project or programme component or the expected cost levels. Drivers are the 'moving parts' of a model and should be identified and set out within the model Specification. It is important to identify relevant drivers, considering the materiality of the cost category, as well as what level of detail or granularity is appropriate for the analysis. Having very granular drivers might be appropriate for a material cost element, and investing the additional time to identify lower-level drivers may add value to the model once built. When identifying drivers, it is recommended to engage a broad range of subject matter experts as well as individuals who have a knowledge of available data to help identify appropriate drivers, as well as the data sets required to model with them.</p>
<b>Risk &amp; Uncertainty</b>	<p>Within the model Specification, areas of risk and uncertainty should be highlighted, including how they will be addressed or their impact assessed by the model (e.g. sensitivity analysis, Optimism Bias, etc.). Where appropriate, ranges should be considered and modelled. See <a href="#">Aqua Book</a> and <a href="#">Green Book</a> for a fuller discussion on how to address risk and uncertainty.</p>
<b>Sign Off</b>	<p>It is good practice to have the model Specification approved and signed off by the Model SRO before starting to build the model. The process can be supported by formalised checklists (see Section 10).</p>

## 5.3 QA, Data and Delivery Plans

- 5.3.1 The initial QA, Data and Delivery Plans, which will have been prepared during the model planning phase, should be reviewed and updated during the design phase to reflect any required changes. For example, the specification of advanced features or requirement for VBA may not have been envisaged when the QA Plan

was initially prepared and this could impact on testing and resource requirements as well as data requirements and delivery timescales.

- 5.3.2 As the level of detail is evolved during the design phase it is important that all data requirements are captured within the Data Plan. A Data Plan helps to mitigate the risk that a model requires unavailable data by codifying the requirement and enabling discussion around it. It should contain key information, such as the time period the data needs to cover, the format it is required in, where it will be sourced from, handling and storage requirements, processing needs, maturity and quality levels, when it is required by as well as who is going to provide it and undertake any pre-processing. Significantly, it should give due consideration to any key associated risks (e.g. around data availability). In time, as model development progresses, the Data Plan will evolve into a Book of Assumptions / Data Log that provides a record of the data underpinning the model (see Section 6.9).
- 5.3.3 The preparation and sharing of QA, Data and Delivery Plans enables discussion and commitment on aspects such as resources and timelines, raises awareness of delivery risks, and provides a means of formalising agreement between model stakeholders. They should be maintained and updated throughout model development (e.g. owing to model circumstance changes).

## 6. Should Cost Model Development

### 6.1 Overview of Development

6.1.1 SCM development is the practice of building the model, populating it with data and undertaking developer testing prior to formal QA and testing.

6.1.2 Investing time in properly planning and designing a model prior to developing it will reduce development risk by, for example:

- Helping to reduce the number of development changes or iterations once model build has started; and
- Helping to ensure a model is designed that is both fit for purpose and efficient in its design.

6.1.3 Following good practice model development approaches will both help to reduce the risk of model construction errors and aid their identification (see SCM Technical Build Guidance). Designing calculations in a transparent and logical way, whilst minimising any unnecessary complexity, will aid model QA and testing and help to reduce the risk of usage errors.

### 6.2 Build to Good Practice

6.2.1 The SCM Technical Build Guidance has been produced to inform the technical design and development of Should Cost Models. It presents good practice principles that helps to manage risk, increase transparency, aid QA and testing, and enhance usability of the model once developed. In summary, the design of a model should be:

- **Planned** – SCM development should follow a structured approach that begins with progressively detailed up-front planning;
- **Logical** – the design of an SCM should be akin to a book, with logic flowing from top to bottom, left to right and front to back;
- **Aligned** – the SCM should be structurally aligned and presented in a consistent manner throughout;
- **Separated** – the SCM should be arranged in modular fashion and there should be clear delineation between inputs, calculations and outputs;
- **Transparent** – the SCM should be intuitive to use and provide transparency over calculations and logic, with no hidden data or calculations;
- **Integrative** – the SCM should be accurate and provide insight in an unbiased, easy to understand and transparent way. There should be no manipulation of the calculation logic to produce a desired result; and
- **Checked** – QA should be integral to the SCM and span the entire lifecycle, from up-front QA planning through in-model checks to formal QA and testing.

6.2.2 The SCM Technical Build Guidance provides the model developer with guidance on how to approach the build of the model, with particular focus on:

- **Workbook Considerations** – including organisation, features and protocols;
- **Worksheet Structure** – including sheet design, principles and presentation;
- **Input Sheets** – including input data arrangement and source referencing;
- **Calculation Sheets** – including layout, principles and formula construction;
- **Formulas and Functions** – including those which should be avoided;
- **Output Sheets** – including usage restrictions and end user communication;
- **Checking and Control** – including in-model error checks and overall QA; and
- **Other Governance Elements** – including documentation and version control.

## 6.3 Data Population

6.3.1 As the model is developed, data will be incorporated into it. This data may not be the final dataset used by the model in producing its analysis, but rather data used to test the mathematical operation during development. Regardless of the data that is being consumed by the model, a Book of Assumptions / Data Log (drawing on the Data Plan) should be updated and maintained to track the data provenance within the model. This will help QA and testing of the model, and provide users with a complete record of the data used by the model. Notably, it may not be possible to perform effective QA and testing on a model in the absence of real data.

## 6.4 Integral Model Checks

6.4.1 The SCM should contain inbuilt checks that should be overtly visible to users of the model. When considering what checks to apply the developer should have regard to areas where models commonly experience error or where calculation logic is at risk of error. These include, but are not limited to:

- **Arithmetic** – do the numbers add up to the correct value;
- **Outlying balances** – are numbers outside an expected range;
- **Incorrect input types** – such as using letters rather than numbers; and
- **Completeness check** – have all expected inputs been completed.

6.4.2 These checks, when written into the model, serve to assist the developer in performing some aspects of self-testing. They also operate to alert the user community to improper use of the model. They can provide confidence by showing that the model continues to operate within the bounds of the checks once in use, and highlight to the user when an erroneous input into the model has been made or an erroneous output generated. They can help manage design and error risk, by confirming that calculations operate within the bounds of checks and have not been corrupted through improper use.

## 6.5 Developer Testing

6.5.1 As the model is developed, testing should be performed routinely by the developer as part of the model build process. Formulae should be checked for consistency against the calculation list contained within the model Specification (see Section 4) to ensure they achieve the desired results. The developer should test the calculation logic and consistency of formulae as they are developed and provide assurances that checks have been performed prior to releasing the model for formal QA and testing. Developer testing should extend beyond the correctness of formulae and it is recommended that the developer runs a range of different data sets through the model to test that the calculation logic operates as intended and produces outputs that are mathematically sound. Checklists are a useful way to structure the developer testing of a model and are discussed in Section 10.

## 6.6 User Guide

6.6.1 A User Guide should be developed for the model. Its purpose is to educate users on the model's purpose, limitations, and how to operate it. It should be as simple and as straightforward as is practicable, but should cover how to interact with the model in several ways:

- **Importing/updating data** – the guide should inform the user how to correctly bring data into the model, where it needs to be placed and any procedures to follow to remove data from previous iterations;
- **Running the model** – the guide should educate the user on how to operate the model, including how to run sensitivities or scenarios and produce outputs; and
- **Interpreting the outputs** – the guide should reach beyond model operators to those responsible for, or using, model outputs. It should reinforce the uses and limitations set out in the model Scope and Specification and provide appropriate guidance on how to interpret outputs.

6.6.2 The User Guide will benefit and should be included as part of formal QA and testing. Its key purpose, however, is to provide instruction to users of the model once it has been released and is in use. User Guides help to manage the risk that models are misused once released and help to reduce key person dependency.

---

“Investing time in properly planning and designing a model prior to developing it will reduce development risk.”

---



## 6.7 Technical or Developer Guide

- 6.7.1 Whilst most models would benefit from a Technical or Developer Guide, it is particularly important and may be a requirement for some models. For example, models that:
- May be or are expected to be iterated or changed over time;
  - Are expected to be used or required for an extended period;
  - Are complex or contain advanced features, such as VBA;
  - Are used to support key or business critical decisions; and
  - Are externally procured or where access to the original developer of the model at a future point in time is considered to be less likely.
- 6.7.2 A Technical or Developer Guide is different from a User Guide. It is focussed on explaining technical aspects of a model to developers to support maintenance or change and also has utility in the context of QA and testing. It should cover the model's construction and how different elements function and interact with each other (the elements that are the most complex or atypical requiring the greatest clarification). Technical or Developer Guides might also cover how to modify or expand the model. Even the requirement for a simple change to a model may render it defunct if it cannot be reliably achieved.
- 6.7.3 Changes to models after they have been developed can significantly raise the risk of errors. This risk typically increases if changes are separated in time from the original development activity and is likely to be even higher if changes are undertaken by a different developer. The availability of a Technical or Developer Guide can help to reduce this risk although, regardless, a modified model should be resubmitted for formal QA and testing following any changes.

## 6.8 Re-use of Previously Developed models

- 6.8.1 User Guides and Technical or Developer Guides are useful tools if a model that has been developed previously is to be used to support a further decision. It is important to complete model scoping and other model planning activities, such as QA planning, and to assess how closely aligned the model targeted for re-use is to the requirements (see Section 8.5). Where previously developed models are not closely aligned to the current requirements, consider whether investing time and effort in developing a new model rather than in reconfiguring the model may be more appropriate. Where the existing model is closely aligned to the requirements, re-use may be appropriate.

## 6.9 Book of Assumptions / Data Log

- 6.9.1 As highlighted above (see Section 5.3), preparing a Data Plan, which will evolve into a Book of Assumptions / Data Log as a model is developed, is good practice.
- 6.9.2 As data can be obtained from different sources, at different levels of detail and with various limitations or constraints, the absence of a Book of Assumptions / Data Log may render it difficult or even impossible to establish data provenance. A Book of

Assumptions / Data Log is a detailed list of all of the data and assumptions that a model uses and their salient characteristics. It might contain information such as:

- **Reference No.** – to link the entry to model inputs;
- **Area** – area the data relates to (e.g. wage inflation);
- **Input Label** – the data label used in the model;
- **Description** – a description of what the data is;
- **Rationale** – why data/ data source is being used;
- **Assumptions** – what’s included (e.g. inc. VAT)
- **Units** - the unit of measure (e.g. £k, \$m, #)
- **Base Year** – Economic Conditions (if appropriate);
- **Filename/Path** – details of source path or files;
- **Source/Provider** – source name or data provider;
- **Date Produced** – how current or up-to-date the data is;
- **Date Provided** – date when the data was provided;
- **Owner** – details of who owns the data/assumption;
- **Review Date** – date when data was last reviewed;
- **Refresh** – the date when data should be updated;
- **Validity** – details of any known data validity periods;
- **Issues** – details of any known issues data issues;
- **Manipulation** – details of any pre-processing applied;
- **Maturity Level** – assessment of data readiness level; and
- **Maturity Plan** – how the data will be matured over time.

6.9.3 The model’s Book of Assumptions / Data Log has several benefits to the end user. It provides a readily accessible overview of the data that the model uses as well as its provenance. It allows for review and challenge of the model’s source data and input assumptions and, together with reference to the model Specification, how it is used within the model to derive outputs. It helps to manage the risk that a model and its source data become disconnected and model results cannot be explained. As well as enhancing the user community’s understanding of the model, the Book of Assumptions / Data Log helps drive auditability of the model and traceability into its dependent data.

6.9.4 The time demands on model developers and quality assurers when undertaking formal QA and testing will increase in the absence of a Book of Assumptions / Data Log. Such demands could substantially delay the model’s delivery and, in-turn, ability to meet its purpose.

## 6.10 Development Sign-Off

6.10.1 Sign-off that the model has been designed in a way that will meet the needs of the user community should have been obtained prior to commencing model build. However, once built and before the model is subject to formal QA and testing, it is

good practice to obtain user sign-off (e.g. via User Acceptance Testing). Recirculating the draft model provides an opportunity to accommodate feedback prior to formal QA and testing. This can help to reduce the impact of change, including the risk that last minute changes are made without adequate time available to properly test them.

- 6.10.2 The model developer should confirm that the model has been built to meet the model Specification and that it has been subjected to appropriate self-testing prior to submitting for formal QA and testing. Preparing a checklist (see Section 10.8) that the developer can complete is one way to help ensure that checks are sufficiently comprehensive and to document the self-testing activities performed as part of model development.

## 7. Quality Assuring and Testing

### 7.1 Overview of Quality Assurance and Testing

- 7.1.1 Whilst the model development lifecycle contains a Review and Formal QA phase, QA should not be seen as a one-off activity and should be undertaken throughout the model lifecycle (see Figure 2). A QA Plan covering testing, other types of QA and controls should be developed as part of initial model planning. This should be revisited periodically throughout the model's lifecycle and especially following any changes to the model's scope or use. Up-front planning helps to ensure that sufficient time and appropriate resource is available to undertake QA and testing and it is not left until the last moment.
- 7.1.2 The activities taken to assure a model should be proportionate to both the criticality of the decision or issue that the model supports and the risk presented by the model. Models that are more complex or sophisticated generally have a higher likelihood of error and present a higher risk. Models that support critical decisions and/or models that are more complex/sophisticated should be subject to more rigorous QA and testing.
- 7.1.3 The IMA Tool will suggest QA and testing activities based on the model's level of criticality and sophistication (see Section 3.2.2). Appendix I: Quality Assurance to this document gives an overview of the different types of QA and testing activities suggested by the IMA Tool and procedural level guidance for the performance of these tests is provided in the SCM Testing Procedures produced by the Sourcing Programme. The [Aqua Book](#) and the [Review of Quality Assurance of Government Models](#) provide commentary on QA and testing within government, more broadly.
- 7.1.4 Notably, quality assurers should be suitably qualified and experienced to discharge their responsibilities. The [Review of Quality Assurance of Government Models](#) notes that there is no substitute for expertise and experience. In general, the skills and experience of many of those supporting quality assurance should be at a level that is equal to or greater than those of the model developer. The effective performance of quality assurance and testing is essential and it should not be viewed as a task for junior resources alone.

### 7.2 Verification and Validation

- 7.2.1 Quality Assurance is more than checking the analysis for errors and alignment with its specification (verification). It must also include checks that the analysis is appropriate, i.e. fit for the purpose for which it is being used (validation). The [Aqua Book](#) contains a fuller discussion on verification and validation of Government Analysis, which is applicable to SCMs. As set out below, the performance of model testing involves three key steps; review, documentation and sign-off:
- 7.2.2 **Review** – this will include the application of an agreed set of test procedures to the SCM and its associated artefacts. The applicable tests, informed by the Initial Model Assessment, should be set out in the QA Plan. The performance of each of

the tests should be undertaken by personnel who are suitably qualified and experienced to do so. The time allocated to undertake each of the tests should be sufficient for them to be performed adequately and should include an allowance for multiple 'review cycles'. These may be required where model developers have not appropriately resolved issues raised by the quality assurer or have introduced new issues as a result of changes. Notably, given the prevalence of errors within models, allowing for a single 'review cycle' is unlikely to be sufficient.

- 7.2.3 **Documentation** – QA Reports should include details of the tests performed and their associated results. Supporting Test Memos should be sufficiently detailed to enable each of the tests to be replicated by an independent person. Copies of the reviewed SCM; any subsequent versions of the SCM as a result of 'review cycles'; all formal correspondence between the model developer and quality assurers; all supporting artefacts (e.g. the model Scope, Specification, User Guide and Book of Assumptions / Data Log) and any subsequent versions; together with details of all of the tests performed and their associated results should be stored in line with file management procedures (see Section 9).
- 7.2.4 **Sign-Off** – The model's QA and testing should be formally signed off prior to using the model or its outputs. Notably, some issues identified during QA and testing may not have been fully resolved and, in turn, may pose potential risks. It is the responsibility of those signing-off the model, and ultimately the Model Senior Responsible Owner (Model SRO), to decide if any unresolved issues need to be addressed before the model is put to use. All unresolved issues should be made transparent in QA Reports and, where appropriate, referenced within the model itself. Plans to rectify unresolved issues should be put in place and reviewed regularly as part of ongoing QA activities.

---

“Quality Assurance is more than checking the analysis for errors and alignment with its specification. It must also include checks that the analysis is appropriate.”

---

## 8. Should Cost Model Use

### 8.1 Governance

8.1.1 Developing appropriate governance and guidance to support the SCM when it is in use will help users of the model to operate it, provide a framework to manage updates and changes, and help to ensure that the model remains fit for purpose. Once a model is released and is in use, mechanisms to track performance and ongoing QA and testing should be considered. Notably, model documentation, such as the model Specification and Book of Assumptions / Data Log, should be updated to reflect any changes to the model and appropriate QA and testing of those changes carried out.

### 8.2 Handover

8.2.1 This activity refers to transitioning the model from those who have developed it, to those who are tasked with using it and informing a decision. When approaching a handover, the developer should consider what the end users will need to know in order to operate the model appropriately. This operating information should be contained within the model User Guide. However, as part of a handover, due consideration should be given to practically demonstrating to key users how to, for example:

- **Populate the model** – how to clear existing data sets and update the model with new data sets;
- **Manage change** – how to update the version control log and, if applicable, configuration control log, within the model (see Sections 10.4 and 10.5);
- **Run scenarios** – if the model has the ability to run scenarios, sensitivities or change analysis parameters, demonstrating how to operate the model (e.g. change inputs or toggle switches); and
- **Configure outputs** – if outputs can be configured (e.g. have a different time series in a chart) showing individuals who are responsible for producing the outputs how to configure them.

### 8.3 Tracking

8.3.1 SCM's may be used to inform decisions that can have impacts spanning several years. When a model is in use, considering how the model will interact with key financial activities, such as long-range planning, budgeting and forecasting, is an important activity. Models can be used to inform financial planning activities and how to track performance against the model's assumptions should be addressed when entering the use phase. Similarly, whether it is appropriate to recalibrate a model to update its analysis for known changes in the environment is a key consideration when a model enters use.

## 8.4 Stakeholder Mapping

- 8.4.1 The model Scope will identify stakeholders who are directly involved either in the design and development of a model or the decision it is designed to support. Example roles and responsibilities are set out in Appendix II: Roles & Responsibilities. When a model enters use and a decision is taken, the stakeholder population may extend beyond these individuals and impact people more broadly (e.g. those in Finance or Commercial). Identifying and communicating with these stakeholders and considering whether to make the model, or parts of it, available to these connected stakeholders is recommended to increase transparency and facilitate performance tracking.

## 8.5 Revisions, Re-Use and Sign-Off

- 8.5.1 The type of decision a model is designed to support together with details of what the model will and will not be able to do will be set out in the model Scope and Specification. Whilst many procurements will require the development of a bespoke model, there will be situations when an existing model can be used to support a decision of a similar type (see Section 3.3). In these situations, the model may require re-configuration and will almost certainly require new data to be able to support the new decision.
- 8.5.2 When investigating whether to reconfigure a model or to build a new one, consideration should be given to potential design compromises, the time to complete in either scenario, as well as issues of resource availability (will there be sufficient resources to oversee the design, development, QA and testing of the model). The scenario with the lowest time to complete will often be preferred.
- 8.5.3 If the decision is taken to reconfigure a model, the risk that it ceases to operate as initially intended should be managed. Developing additional functionality as well as altering the model (e.g. to allow more categories of cost) carries an inherent risk that the model becomes corrupted and does not provide insight at the required level of accuracy.
- 8.5.4 If revising a model, it is important that the model development lifecycle is followed. Copies of the model Scope and Specification (inc. Design) should be reviewed, revised as appropriate and approved, including any associated plans (e.g. the Data, QA and Delivery Plans). Any changes to the model should be made in line with the SCM Technical Build Guidance and subject to developer testing ahead of formal QA and testing. Even though the model will have undergone QA and testing prior to initially entering use, it should be subject to formal QA and testing in order to address the new data set, check for errors and check that changes have been made in a way that achieves the new purpose.
- 8.5.5 Notably, it is important to consider whether any changes to the model and/or its use have increased its risk profile (see Section 3.2.2 for Initial Model Assessment). For example, if a model has been changed to include a new type of advanced analysis, the QA Plan that was used when the model was initially released may not be appropriate to use for the next iteration, and a more comprehensive set of tests may need to be performed. Particular attention should be paid to any limitations to

the original model or recommendations made by quality assurers that have not been implemented in the original model (see Section 7.2.4).

- 8.5.6 Making revisions to a model can carry greater risk than developing a model from scratch, especially if there is no developer continuity and/or no Technical or Developer Guide (see Section 6.7). This is the reason to follow the same development lifecycle approach and to structure the reconfiguration of a model around the same good practice risk management approaches employed during its initial development.

## 8.6 Periodic Testing

**Separate to the tracking of a model's performance, it is good practice to have the model tested periodically.**

The nature of the periodic testing should be governed by the principle of proportionality and have regard to the usage of the model. Models that are more critical or have been used extensively and had multiple updates to the data should be tested more frequently and thoroughly than those which have not. It is important to approach periodic testing through the dual aspects of verification (that the model operates properly) and validation (that the model is appropriate).



## 9. File Management

### 9.1 Overview

9.1.1 Models may be critical to the business or contain sensitive data, meaning that they can only be accessed by a limited number of individuals. The development of models will also involve multiple files, such as the model Specification, Book of Assumptions / Data Log, source data files, QA Reports and versions of the model itself. All of these should be appropriately managed. Thus far this guidance has focussed on how to approach the practice of developing a model in a way that manages risk and drives efficiency. This section focusses on considerations in relation to managing the risks associated with multiple files and their access.

### 9.2 File Management Considerations

9.2.1 **Sensitivity Control/ Protective Markings** – consideration should be given to the sensitivity of files used in the production of a model, as well as the model itself. Where a protective marking is required all files should be appropriately classified and treated in line with relevant guidelines for handling sensitive information (see [Government Security Classifications](#) and [Guide to GDPR](#)).

9.2.2 **Version Control** – models, datasets and even planning and QA documents will be iterated. It is good practice to agree and maintain a version control system for managing files. As part of this, version control logs (see Section 10.4) can be used to provide a record of the version number, changes from the previous version and details of when and who made and approved the changes.

9.2.3 **File Locations** – files should be stored in an agreed location and the sharing of files via email should be avoided. Within the location, files should be arranged within a logical folder structure that keeps planning, data and model files separate. Equally, previous versions or drafts should be stored in a dedicated folder. Locating files in such a way increases auditability of the model, assists with access permissions through applying access restrictions on sensitive folders, and reduces the risk of an incorrect version being used.

9.2.4 **Access Controls** – all files should be stored in secure shared areas that are only accessible to relevant individuals and with the appropriate read or write access. If files are stored in areas with wider access, they should be password protected to reduce the risk of unauthorised access and/or change (see Password Usage below).

9.2.5 **Protection** – it is good practice to apply cell and worksheet protection to spreadsheet-based SCMs to reduce the risk of inadvertent corruption (see SCM Technical Build Guidance).

9.2.6 **Password Usage** – passwords can be used to add an extra layer of security to a model (subject to contracting authority policies). Worksheet and workbook level protection, supported by passwords, can be used to help manage the risk that files

are corrupted through misuse or incorrect data entry. Where files contain sensitive information (e.g. Personal Identifying Information) consider applying encryption at the file level. However, the sufficiency of this needs to be carefully considered and appropriate advice should be sought.

- 9.2.7 **Password Management** – any use of passwords should be in-line with contracting authority policies. If they are used it is important that passwords are not lost or forgotten as their absence can render a model unusable. Maintaining records in a centrally managed vault and using secure strings can reduce associated risks.
- 9.2.8 **File Backups** – the sufficiency of backups afforded by the contracting authority’s IT system should be explicitly considered and supplementary measures put in place as appropriate. If organising backups on shared drives, they should be within separate folders and appropriately version and access controlled.
- 9.2.9 **File Archiving** – the requirement and process for storing copies of historical versions used to support certain key decisions should be considered for all SCMs. Particular thought should be given to the implications of overwriting historical data, the need or otherwise to break data links, the need to save files as read-only, and the implications of software compatibility in the future.
- 9.2.10 **File Sharing** – the need for specific controls in relation to the sharing of SCMs should be considered on a case-by-case basis. Particular thought should be given to data sensitivity, impact on version management and the use of a Non-Disclosure Agreement (NDA) or other measures if 3<sup>rd</sup> parties are involved.
- 9.2.11 **Inventory** – some SCMs may be classified as Business Critical Models (BCMs) and may need to be added to associated inventories. These records should include information such as the names of the model developer, quality assurers and Model SRO, the SCM’s development status, current QA status and planned QA and testing activities.
- 9.2.12 **Periodic Review** – the management of an SCM and its associated files should not be seen as a one-off event. Certain activities should be undertaken at regular intervals and may also be triggered by events. These include retesting the SCM and reviewing file management procedures, including file access rights and file retention and backup protocols. They may also extend to periodically testing the ability to restore files from archive or backup.

## 10. Tools, Templates & Checklists

### 10.1 Overview

- 10.1.1 Tools, Templates and Checklists have considerable utility in the development of SCMs.
- 10.1.2 Tools can be used to augment a model's functionality (such as spreadsheet add-ins to perform risk and uncertainty analysis) or automate elements of the model's review (e.g. tools that support Verification or Analytical Review).
- 10.1.3 Templates can be used to help structure model development, reinforce good practice and reduce the time taken to develop a model.
- 10.1.4 Checklists provide structure and help to formalise requirements. They can be useful communication devices as models transition between development stages, informing the recipient of the model what checks or activities have been performed at the previous stage. They can be used to support those responsible for overseeing model development (see Section 2.4) and drive benefits across the SCM's lifecycle:
- **Efficiency** – by codifying and structuring the process and streamlining review;
  - **Effectiveness** – helping to reinforce good practice and limiting rework;
  - **Consistency** – driving completeness and uniformity of approach; and
  - **Audit** – providing a record of activities undertaken and their sign-off.

### 10.2 Planning and Design Templates

- 10.2.1 Templates for model planning and design, including delivery and QA planning templates, provide structure and help to drive completeness and consistency of approach. They can bring efficiencies and help to reinforce adherence to good practice. For example, they can help to ensure that resource planning extends beyond model developers and includes other roles (e.g. quality assurers), and that resource requirements also consider the more operational facing roles, such as Model SROs and customers.
- 10.2.2 The Sourcing Programme have produced a number of planning and design templates that can be adapted as required to support the production of SCMs. These include the SCM Scoping Template, Planning Template, QA Plan Template and Specification Template Example. The SCM Build Template also includes a Book of Assumptions / Data Log that that can be adapted to support preparation of a Data Plan.

## 10.3 Model Build Template

10.3.1 Model build templates, which can be used to inform and form the basis for development of an SCM, have many benefits, including:

- **Compliance** – reinforces and helps drive compliance with good practice;
- **Effort** – reduces the time to create, develop and educate users on a model;
- **Consistency** – drives consistency of styles and ease of understanding, and
- **Features** – includes good practice features (e.g. error check network).

10.3.2 The Sourcing Programme have produced an SCM Build Template that can be adapted as required to form the basis for development of an SCM.

## 10.4 Version Control Log

10.4.1 Version control logs help to track changes and updates to a model over time. It is recommended that they are maintained for all SCMs. As models are iterative, with structural and particularly data changes and updates being commonplace, it is important to maintain version control. This provides the user population with auditability of changes and a mechanism to help ensure that the SCM they are using is the appropriate version.

10.4.2 The SCM Build Template, produced by the Sourcing Programme, includes a Version Control Log that that can be adapted and/or used within the SCM Build Template or another workbook as required.

## 10.5 Configuration Control Log

10.5.1 Configuration control logs help to track different model configurations, such as sub-option combinations. Whilst it may be possible to accommodate configuration control within version control logs, the need for a dedicated configuration control log should be considered. This may be particularly pertinent for SCMs that can be configured in a wide range of manners. Configuration control provides the user population with auditability of model configuration and a mechanism to help ensure that the SCM they are using is appropriately configured.

## 10.6 Review Tools

10.6.1 Automated tools that support the testing of SCMs (e.g. Logic Testing and Analytical Review) are a key component of an effective risk management approach. Their use not only increases the effectiveness and reduces the time taken to undertaken specific tests, but they enable the performance of tests that could not otherwise be undertaken. Use of these tools, which is recommended for more rigorous forms of testing, may require broader considerations, such as their compatibility with the contracting authority's IT policy, how users will be trained on their use and how testing will be performed and documented.

10.6.2 There are several providers of Logic Testing tools, which may also be known as Model Auditing or Verification tools. In general, they produce model maps (a diagrammatic representation of a model showing its elements and their relationship to adjacent elements), unique formula listings and other reports such as VBA, Named Range and External Link listings, that can help facilitate a model's testing. They can be purchased, downloaded and installed from a provider's website.

## 10.7 Good Practice Checklist

10.7.1 Good practice checklists provide an efficient and structured approach to confirming an SCM has been built in line with good practice (e.g., the SCM Technical Build Guidance). They are not a substitute for other forms of QA and testing but should be used to drive completeness and assess compliance with the principles.

10.7.2 The Sourcing Programme have produced an SCM Good Practice Build Toolkit that can be used to structure and support the performance of a Good Practice Critique.

## 10.8 Development Checklist

10.8.1 A checklist that summarises and provides confirmation that specific activities, such as developer testing, have been performed is a recommended component of an effective risk management approach. It supports the communication of activities amongst the SCM stakeholder community, helps to reinforce good practice and provides a reference point for areas of focus from a review standpoint. It helps individuals to autonomously perform QA in an auditable, regimented way, that can be easily documented.

10.8.2 The Sourcing Programme have produced an SCM Development Checklist that can be adapted as required and used across the model development lifecycle.

---

“Good practice templates provide an efficient and structured approach to confirming an SCM has been built in line with the good practice.”

---

# 11. Appendix I: Quality Assurance and Testing

This appendix describes a number of QA and testing activities, including those that may be suggested by the Initial Model Assessment (see Section 3.2.2). It covers both the activity itself and the key outputs from it.

Activity	Description
<b>Delivery Plan</b>	The Delivery Plan documents delivery timelines, delivery risks and mitigations, and resource requirements. It will be informed by and should be prepared during the model planning phase as part of producing the model Scope. It should be maintained and updated throughout model development (e.g. owing to model circumstance changes). The preparation and sharing of a Delivery Plan supports and formalises agreement between model stakeholders. It enables discussion and commitment on resources and timelines, and raises awareness of delivery risks.
<b>QA Plan</b>	The QA Plan documents planned QA and testing activities over the model's lifecycle. It will be informed by the Initial Model Assessment and should include activities that are considered to be appropriate for the model based on its risk profile. It should be prepared during the model planning phase as part of producing the model Scope. The QA Plan supports and formalises agreement between model stakeholders. It enables discussion and commitment on the level of Quality Assurance and testing that needs to be performed in order for the user community to have confidence in the model outputs.
<b>Data Plan</b>	The Data Plan documents the data sets that are going to be used in the model. A preliminary Data Plan will be informed by and should be prepared during the model planning phase as part of producing the model Scope. It provides initial visibility of what data is required and its expected source, supports its collection and highlights where data is deficient or sources need to be identified. The preliminary Data Plan should be updated to reflect the agreed model inputs set out in the model Specification. At this point it should extend beyond data sources, to cover data access, storage, processing, formatting and any applicable handling requirements; timelines; roles and responsibilities; and risks and mitigations. The Data Plan should be maintained throughout model development. It will inform the Book of Assumptions / Data Log, which provides provenance to all of the data and assumptions that underpin the model.
<b>In Model Checks</b>	This is a form of 'self-testing' performed by the model itself. Here, automated checks are designed into and throughout the model by the developer during the model's construction. These checks are not solely focussed on arithmetic correctness and can extend much wider. For example, checks on whether inputs are complete or current, if input or output values fall within expected ranges, if the model's name aligns with the Version Control Log, etc. The error checks should feed into a network that overtly informs the model operator of the presence of potential errors anywhere within the model (see SCM Technical Build Guidance). The in-model checks should not be seen as a substitute for developer testing or formal QA and testing but rather an individual aspect of overall QA.
<b>Version Control Log</b>	This is a dedicated log, based on a structured methodology, that highlights the current version of the model and changes made versus previous iterations. It should

Activity	Description
	<p>be maintained to provide an audit trail of changes together with their rationale and accountability. It supports model QA and testing, can aid version roll-back and helps to mitigate the risk of using an incorrect model version. Effective version control also helps to alert the user community to changes that have been made as the model is developed or used.</p>
<b>Developer Testing</b>	<p>Having the model tested by the developer prior to formal QA and testing helps to reduce the risk of error and streamline the review cycle. Prior to release for formal QA and testing, the developer should undertake and document a self-review of all aspects of the model and any associated documentation. This should include checking that the design logic is correct and aligns with the model Specification; that the model is constructed in line with good practice (see SCM Technical Build Guidance); that the in-built error checks operate correctly; and that the model is able to run all scenarios/ sensitivities and functions appropriately. Codifying tests in a Checklist will provide a simple and regimented way for a developer to confirm that a model has been developed in line with the SCM Technical Build Guidance to meet the model Specification and that appropriate developer testing has been performed prior to release for formal QA and testing.</p>
<b>User Acceptance Testing</b>	<p>This is the process of having a model tested by the end user 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. For example, that inputs can be entered into the model, outputs can be accessed and model switches are operational. Whilst User Acceptance Testing (UAT) may expose issues with a model it is not focussed on the identification of errors and should be seen as a compliment to and not a substitute for formal QA and testing.</p>
<b>Verification</b>	<p>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 inc. Design). It involves the independent application of a broad range of analytical techniques that include the review of:</p> <ul style="list-style-type: none"> <li>• all unique formulas on a cell-by-cell basis;</li> <li>• a Structural Review (model maps);</li> <li>• all VBA code on a line-by-line basis;</li> <li>• all formulas for inconsistencies;</li> <li>• all Named Ranges for errors;</li> <li>• all external links for appropriateness;</li> <li>• all inter and intra-sheet data flows;</li> <li>• a Good Practice Critique; and</li> <li>• the overall model logic and design.</li> </ul> <p>Performing a Verification requires specialist skills and software tools and can be labour intensive. 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 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 Parallel Build (see below).</p> <p>The quality assurer does not necessarily need to be from a 3rd party provider but should be unaffiliated with the development team and suitably qualified and experienced to undertake the review. The output of Verification is a documented list of tests performed and results against them. Notably, Verification is unlikely to be sufficient to confirm, or otherwise, the fitness for purpose of a model and additional tests may also be required (e.g. Validation)</p>

Activity	Description
<b>Validation</b>	<p>This 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 inc. 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 focusses on assessing the viability of the data and assumptions, the way they are manipulated by the model (in the context of its intended purpose), and the authenticity of model outputs. It will include Commercial Review and Analytical Review as well as a more granular review of model input data and assumptions and model methodology.</p> <p>Notably, depending on data volumes, a statistically significant data sample may be employed over full sampling.</p> <p>The output of Validation includes an assessment of the completeness of the Book of Assumptions / Data Log, an assessment of the maturity and appropriate use of data, an assessment of the model's overall logic against its intended purpose together with the results of both the Commercial and Analytical Review. Notably, Validation is unlikely to be sufficient to confirm, or otherwise, the fitness for purpose of a model and additional tests may also be required (e.g. Verification).</p>
<b>Structural Review (Model Maps)</b>	<p>This is a review performed to check that the model's structure is in line with good practice approaches (e.g. logical, aligned, separated). A Structural Review invariably requires the application of specialist testing software in order to check model design, data flows, and formula consistency in an efficient and thorough manner. The output is a report highlighting examples of where the model's structure falls short of good practice together with any identified formula inconsistencies. Notably, it is driven top down and focuses solely on structure and apparent structural anomalies.</p> <p>A Structural Review can be included within Verification, but can also be performed when other Verification tests are not being undertaken.</p> <p>A Structural Review, which can identify some common errors, takes far less time than many other Verification tests and may be appropriate for low-risk models or as part of other tests (e.g. Developer Testing or Peer Review).</p> <p>Elements of a Structural Review may be included as part of a Good Practice Critique.</p>
<b>Good Practice Critique</b>	<p>This is an independent quality assurer's assessment of the model's compliance with good practice (see SCM Technical Build Guidance). Model's that are built in line with good practice generally have a lower risk of errors, including inherent errors and errors resulting from misuse. The output is a report highlighting the extent to which the model adheres to good practice and examples of where it falls short. The output can be used to help inform an assessment of developer proficiency or, more usually, a model's overall risk. This, in-turn, may inform a need for remedial action and/or more stringent QA and testing activities. Notably, a Good Practice Critique will not cover every aspect of the model's adherence to good practice and, moreover, is not generally intended to identify errors. A Good Practice Critique should be included as part Verification but can be performed as a standalone activity to complement other forms of QA and testing.</p>
<b>Analytical Review</b>	<p>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</p>



Activity	Description
	<p>themselves. The effective application of 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). The output includes a report covering the tests performed against expected and actual outputs and associated commentary (e.g. as/not as expected). Analytical Review usefully benefits from graphing and charting and may best be supported by software tools. It should not be seen as a substitute for other forms of QA and testing but rather a specific type of QA in its own right. As a standalone test it can provide insights into a model's behaviour and may provide additional comfort, particularly over models with complex logic. Analytical Review should be included as part of Validation.</p>
<p><b>Partial Verification</b></p>	<p>A Partial Verification will include a reduced breadth and/or depth of testing compared to Verification and correspondingly require less effort (see Verification overview above). It requires clear up-front agreement on what will and will not be performed, which should be documented as a set of agreed upon procedures. It will not provide the same degree of comfort as a Verification and the limitations should be well understood.</p> <p>QA Reports should clearly state that a Partial Verification has been undertaken and highlight key limitations. When agreeing the scope of the Partial Verification, the trade-offs between time to undertake versus benefits should be carefully considered. For example, Structural Review (model maps) and Good Practice Critique generally take less time than other aspects of Verification. Oftentimes, it is the scope of the unique formula review that is reduced owing to the time intensive nature. In these cases, the focus of unique formula review may be limited to specific sheets, or sections of sheets containing complex formula, or those that drive the highest proportion of overall costs. Notably, Verification is unlikely to be sufficient to confirm, or otherwise, the fitness for purpose of a model and additional tests may also be required (e.g. Validation).</p>
<p><b>Partial Validation</b></p>	<p>A Partial Validation will include a reduced breadth and/or depth of testing compared to a Validation and correspondingly require less effort (see Validation overview above). It requires clear up-front agreement on what will and will not be performed, which should be documented as a set of agreed upon procedures. It will not provide the same degree of comfort as a Validation and the limitations should be well understood. QA Reports should clearly state that a Partial Validation has been undertaken and highlight key limitations.</p> <p>When agreeing the scope of the Partial Validation, the trade-offs between time to undertake and versus benefits should be carefully considered. For example, Commercial and Analytical Review generally take less time than other aspects of Validation. Oftentimes, the volume of data tested is reduced owing to the time intensive nature of validating inputs. In these cases, the focus may be on cost drivers with the greatest impact on overall costs and/or sampling approaches used to reduce the volume of data tested. Notably, Validation is unlikely to be sufficient to confirm, or otherwise, the fitness for purpose of a model and additional tests may also be required (e.g. Verification).</p>
<p><b>Peer Review</b></p>	<p>Having a model reviewed by an independent internal or external peer. With this approach the quality assurer will test the model for structural, logical and output integrity. The aim of Peer Review is to independently confirm that a model is appropriate and operates as intended. Those undertaking Peer Review should not be connected to the development of the model and should be separate from pre-testing activities such as preparing the model Scope or Specification. It is important that the tests to be performed by the quality assurer are agreed and documented,</p>

Activity	Description
	with key success factors developed, prior to undertaking testing. The output of the Peer Review is a list of tests performed and outcomes of these tests. When planning Peer Review, it is important to consider what aspects of QA and testing will be covered (e.g. Verification and Validation). Without such a plan, and documentary evidence of tests performed and their results, a Peer Review may offer little or no comfort.
<b>Commercial Review</b>	Having a model reviewed from a commercial perspective. With this approach a commercial specialist with subject matter expertise will test the model for commercial authenticity. The approach is generally more holistic than analytical techniques although may draw upon some of them. It focuses on confirming the appropriateness of the model's overarching logic in the context of the service, project or programme being costed as well as whether input and output values lie within expected ranges. Analytical techniques, such as regression and comparative analysis, and the construction of parallel '10 Line Models' for components of the service, project or programme are generally employed in support of a Commercial Review. The output of the Commercial Review is a list of tests performed or comparisons made and the resultant outcome. A Commercial Review aims to provide a degree of comfort at an order of magnitude level and should not be relied upon to identify errors directly. It is recommended for all models and is complementary to other, more granular, QA and testing approaches. Commercial Review should be included as part of Validation.
<b>Parallel Build</b>	This is a rigorous technique that uses re-computation to test the mathematical operation of a model by producing an identical output. This technique involves the independent development of a parallel model, to the same model Specification, followed by a comparison of results produced. In some situations, it may be considered as preferable to Verification as, for example, it can help mitigate risks of omission and interpretation and provide independence of mind. In other situations, it may be the most viable option. For example, if software tools to support Verification are inaccessible, where there is a desire to replace the model with the parallel model or where 'real-time' parallel development is the only means of 'verifying' the model in the available timescales. For models that require very high levels of comfort it may be also considered in addition to Verification. Notably, Parallel Build in-itself is unlikely to be sufficient to confirm, or otherwise, the fitness for purpose of a model and additional tests may also be required (e.g. Validation).
<b>QA Report</b>	A QA Report should be produced for all models. It specifies the tests and other QA and governance activities that have been performed on the model, the results of testing, and any associated limitations (e.g. breadth or depth of testing, issues that have not been resolved, model use restrictions, etc.). It provides key evidence that appropriate governance, QA and testing has been applied in support of the model's sign-off as being fit for purpose for a specific use. The QA Report will usually be supported by Test Memos that provide sufficient detail to enable the tests to be repeated by an independent person.
<b>Ongoing QA</b>	For models that are to be used over a longer timeframe than the specific procurement, periodic review should be performed. This checks that ongoing control activities have been followed (e.g. maintenance of the Version Control Log and adherence to file management procedures); that any recommended actions (e.g. following previous QA and testing) have been implemented; and, through re-testing, that the model continues to operate as intended and remains fit for purpose. This review is focussed on ensuring that the model is managed and functions as intended and provides a formal mechanism to identify improvements (e.g. amendments required by the user community). Confirmation that controls (including

Activity	Description
	<p>QA and testing procedures) remain appropriate, should also be made. For high-risk models, ongoing control activities may also include an Internal Audit Review or its equivalent. This will focus on the model's current QA status, including whether procedures have been followed (e.g. assignment of a Model SRO, maintenance of QA records). It may also test if recommendations (e.g. Model SRO training) have been implemented, if model outputs align (e.g. Business Case costs match the version of the model that underwent QA and testing) and, through re-testing, the effectiveness of QA or the extent to which the model adheres to good practice. This review is focussed on testing the overall quality and effectiveness of QA, including the application and sufficiency of procedures and controls. The output of the review will be a documented list of tests performed and outcomes against these tests. The results across a number of models may be compiled into a report for senior contracting authority stakeholders.</p>

## 12. Appendix II: Roles & Responsibilities

Role	Typical Responsibilities
<i>Model Developer</i>	<ul style="list-style-type: none"> <li>• Driving production of the model Specification (inc. Design), Data, QA and Delivery Plans, and seeking their approval and sign-off ahead of model development.</li> <li>• Updating the model Scope, Specification (inc. Design) and other documentation to reflect any agreed changes.</li> <li>• Building the SCM in line with requirements and good practice guidance.</li> <li>• Populating the SCM with data for self-testing and to support formal QA and testing, handover and release of the SCM for use.</li> <li>• Producing and updating the Book of Assumptions / Data Log with key information pertaining to the collected data.</li> <li>• Undertaking self-testing of the SCM throughout development and prior to release for formal QA and testing.</li> <li>• Working with Quality Assurers to implement changes required to address issues identified as part of QA and testing processes.</li> <li>• Producing the model User Guide and, if required, handover training materials and the model Technical or Developer Guide.</li> <li>• Undertaking demonstrations and familiarisation sessions as required and producing interim results to support in-flight QA and testing.</li> <li>• Implementing any file management procedures that may be applicable during the development of the SCM.</li> </ul>
<i>Model SRO</i>	<ul style="list-style-type: none"> <li>• Model Senior Responsible Owner who takes overall responsibility for the SCM and its use, including its QA and testing and governance throughout its lifecycle.</li> </ul>
<i>Model Customers</i>	<ul style="list-style-type: none"> <li>• Inputting to the model Scope and Specification and confirming the suitability of the model's design to support decision-making requirements.</li> <li>• Inputting to the Delivery Plan and confirming that the overall timescales are in-line with requirements.</li> </ul>
<i>Model Operator</i>	<ul style="list-style-type: none"> <li>• Undertake familiarisation and/or training as required to operate the model.</li> <li>• Implementing the required file management processes and procedures.</li> <li>• Running the model to produce the required outputs and to interpret the result.</li> <li>• Refreshing the input data as required to run the model and produce the outputs.</li> </ul>
<i>Model Architect</i>	<ul style="list-style-type: none"> <li>• Leading the model design and taking responsibility for the associated model documentation, including agreeing any required changes during development.</li> <li>• Overseeing the model development process and providing technical and design support and challenge during build.</li> </ul>
<i>Data Providers</i>	<ul style="list-style-type: none"> <li>• Inputting to all relevant aspects of the Data Plan including the provisional timescales and risks.</li> <li>• Sourcing the data required by the model and undertaking any data pre-processing that may be required.</li> <li>• Updating key stakeholders on progress and any required changes to the Data and Delivery Plans.</li> </ul>
<i>Quality Assurers</i>	<ul style="list-style-type: none"> <li>• Undertaking QA and testing and producing associated documentation, such as QA Reports and Test Memos.</li> <li>• Liaising with the Model Developer to explain identified issues where further clarity is needed and undertaking re-testing, as required.</li> </ul>

## 13. Appendix III: Glossary & Abbreviations

### Glossary & Abbreviations:

- **BCM** - A 'Business Critical Model' is one that informs and is essential to financial decisions or the achievement of business priorities where errors could result in significant financial damages, legal damages, reputational damages and/or penalties.
- **BCM List** - A list or inventory of all 'Business Critical Models' maintained within the contracting authority. It will include information such as model name, production date, developer name, quality assurer name, Model SRO name, development status, current QA status and planned QA and testing activities.
- **IMA** - An 'Initial Model Assessment' is the process of evaluating the risks associated with an SCM to inform proportional QA and testing activities. It considers both the SCM's criticality or impact, with reference to the decision it is designed to support, together with environmental factors that drive risk such as the SCM's complexity and the nature of underlying data.
- **IMA Tool** - The 'Initial Model Assessment Tool' is a tool provided by Cabinet Office to support the performance of an IMA in a structured and efficient manner (see Section 3.2.2). It provides a Low/Med/High risk score for the SCM based on user supplied answers to questions and suggests QA and testing activities to consider.
- **Model SRO** - The 'Model Senior Responsible Owner' is the single individual who is ultimately accountable for the SCM, including its delivery, outputs and all aspects of its governance, control and use.
- **Monte Carlo** - Monte Carlo simulation is a mathematical technique for the inclusion of risk and uncertainty in an SCM. Its practical application demands automation through specialist software in the form of, for example, a Microsoft Excel add-in or VBA. SCM outputs can be presented as a range of possible outcomes and the probabilities that they will occur.
- **QA** - 'Quality Assurance' is the maintenance of a desired level of quality in an SCM through focussed attention and the application of governance and control activities to every stage in the process of producing, maintaining, using and decommissioning one.
- **SCM** - A 'Should Cost Model' is a type of financial model that evaluates what a service, project or programme 'should' cost over its whole life.
- **SQEP** - 'Suitably Qualified and Experienced Personnel' are individuals or teams of people with sufficient skills, experience and understanding to be relied upon to discharge their duties to the required standards.
- **Test Memo** - a supporting document to the QA Report that sets out test procedures performed on a model along with the results of the tests and in sufficient detail to enable them to be repeated by someone independent, as may be required.

## 14. Appendix IV: Designing a Model

During the model design phase (see Figure 2) the model Scope will be evolved into a more detailed analytical plan. This, as set out in Section 4, includes development of the model Specification (inc. Design) as well as refinement of the materials produced during the planning phase, such as the Data Plan. The table below summarises the key outputs from the model design phase and poses a series of supporting questions.

### 1: Articulate the Model Design

- A schematic of the model that depicts each of its worksheets, their function (e.g. input, calculation, output) and how they interact with other sheets (e.g. intra-sheet data flows).
- Depict how model outputs will be organised within and across different worksheets within the model (e.g. purpose, level of detail, type).
- Summarise key calculation steps performed on each worksheet including how they will be arranged and sequenced.
- Depict how model inputs will be organised within and across different worksheets within the model (e.g. organised by function, nature, form, provider etc.).

#### Key Questions

- How should the model's outputs be arranged?
- How are the model's inputs going to be organised?
- What are the key calculation steps and their arrangement?
- What is the required sequencing of model calculations?
- Does the design of the calculations limit duplication?
- How flexible is the design to accommodate change?

### 2: Detail Advanced Analysis/Specialist Input

- For more sophisticated models the use of advanced analytical techniques may be required. For example, models may require statistical analysis (e.g. Monte Carlo simulation) or accounting approaches (e.g. Net Present Value) and some may require Economic or Financial modelling techniques. These requirements should be captured in the model Specification along with any associated data needs. Where specific or specialist resources are required, they should be reflected in the Delivery Plan, which should have been prepared during the model planning phase.

#### Key Questions

- Is complex modelling and/or the use of advanced techniques essential to produce model outputs?
- What specialist input or advice is needed to achieve the model's objectives?

- What additional data (detail and sources) are required to support the advanced features?
- Who will develop and who will QA and test the advanced features within the model?
- Who will provide specialist input or advice?

### 3: Detail Advanced Model Functionality

- Detail any advanced functionality or automation required by the model. For example, functionality that sits outside of the basic Excel suite (e.g. Excel add-ins) or requires the use of VBA (e.g. Custom Functions).
- Consider who will deliver, QA and test the advanced functionality requirements and whether they are suitably qualified and experienced and have access to any tools required.

#### Key Questions

- Is advanced functionality required or can results be achieved using basic Excel functionality?
- Does providing and quality assuring the functionality require specialist input? If so, who will provide the capability (independence of development and testing is required)?
- If the functionality is delivered through the use of Excel add-ins, are there any IT factors that may impinge on its use?

### 4: Update Data Plan

- Update the Data Plan, which should have been prepared during the model planning phase, to include all input data and assumptions required by the model.
- Consider data availability and data quality and identify areas where obtaining data may be problematic or data quality may be low.
- Where data is not readily available detail actions to source or generate the data (e.g. through benchmarking).
- Consider any data processing requirements that may be required to transform the data for SCM population.
- Once agreed, the Data Plan will form the basis for the initial Book of Assumptions / Data Log. The Book of Assumptions / Data Log should cover each data point within the model and provide salient information pertaining to its provenance, such as:

#### Key Data Information

- **Reference No.** - to link the entry to model inputs;
- **Area** - area the data relates to (e.g. wage inflation);
- **Input Label** - the data label used in the model;
- **Description** - a description of what the data is;
- **Rationale** - why data/ data source is being used;

- **Assumptions** - what's included (e.g. inc. VAT);
- **Units** - the unit of measure (e.g. £k, \$m, #);
- **Base Year - Economic Conditions** (if appropriate);
- **Filename/Path** - details of source path or files;
- **Source/Provider** - source name or data provider;
- **Date Produced** - how current or up-to-date the data is;
- **Date Provided** - date when the data was provided;
- **Owner** - details of who owns the data/assumption;
- **Review Date** - date when data was last reviewed;
- **Refresh** - the date when data should be updated;
- **Validity** - details of any known data validity periods;
- **Issues** - details of any known issues data issues;
- **Manipulation** - details of any pre-processing applied;
- **Maturity Level** - assessment of data readiness level; and
- **Maturity Plan** - how the data will be matured over time.

The Book of Assumptions / Data Log should be maintained and updated as appropriate over the life of the model.



## 5: Document Model Calculation Methodology

- Identify the key calculations required to achieve the model's objectives and any specialist input needed (e.g. finance department for capitalisation).
- Document calculation methodologies in expression as opposed to mathematical form (e.g. Staff Costs = Salary + National Insurance + Pension).
- Calculations should be arranged progressively and logically, showing where inputs will be used e.g.:

Input	Calculation Methodology
Salary	Input
National Insurance	Input
Pension %	Input
Pension	= Salary x Pension %
Staff Costs	= Salary + National Insurance + Pension

- Calculations should be grouped and organised in a logical way, for example to aid end user understanding of model logic.

### Key Questions

- Are all calculations required for the model to operate listed?
- Can calculations be simplified to achieve the target outcome?
- Are the calculations laid out logically and easy to read?

## 6: Prepare Example Outputs and Inputs

- Prepare exemplar model outputs, to include graphs and charts, financial tables or other pro-forma outputs and socialise with end users.
- Target model outputs should be agreed with the user community and changes thereafter should be limited.
- The arrangement of model outputs should be logical and user friendly (e.g. grouped by user category, or by type/function).
- Prepare exemplar model inputs that show the organisation and layout of model inputs together with expected units.
- Cross reference model inputs against the Book of Assumptions / Data Log and the calculation methodology set out to check for completeness / appropriateness.
- Target model inputs should be agreed with the key stakeholders (e.g. data providers and model customers) and changes thereafter appropriately coordinated.

**Key Questions**

- Have draft outputs been agreed with the user community?
- Are additional outputs required or can they be anticipated?
- Have draft inputs been agreed with key stakeholders?
- Have draft inputs been checked for completeness and appropriateness?
- Can the outputs be delivered through the listed inputs and calculations?
- Are there any outputs that are unnecessary and not required?

**7: Approval & Sign Off**

- The model Specification should include a list of stakeholders who are responsible for confirming that the Specification will inform development of a model that is fit for purpose. These may include model customers, users and data providers.
- The Model Senior Responsible Owner (Model SRO) should be included within this list.
- Once the model Specification is prepared, the listed individuals should agree that it is appropriate to deliver the required insight and output (customers/users) and that the required data (data providers) will be available when required and in the right format.
- Following approval, the Model SRO should sign-off the model Specification, allowing development activity to begin.

**Key Questions**

- Who will approve the model outputs as complete?
- Who will approve the calculation logic and methods?
- Who will confirm that the required data are available?
- Who is the Model SRO (responsible for signing off the model Specification ahead of model build)?

## 15. Appendix VI: Cost Estimating Techniques

There are a broad range of approaches that use benchmarks and/or past experience from reasonable comparators to provide robustness and objectivity to cost estimating. For example:

- **Expert Opinion** – This technique involves canvassing opinion from experts in a particular field, to provide insight into expected cost levels. It can provide service, project or programme specific insight and allow for high-level cost estimation when data is deficient, but lacks the evidence base of other techniques and is susceptible to bias;
- **Analogous Estimating** – This technique takes data and learnings from previous services, projects or programmes and uses them to represent what future costs might be. It takes information about elements of services, projects or programmes that are similar in nature and allows them to inform what proposed costs might be in future models. These estimates can then be recalibrated for known differences between services, projects or programmes and used within a model. These techniques are often used in early phases of model development and are less accurate than other methods;
- **Parametric Estimating** – This is a statistical technique that uses historical data relating to key cost drivers to provide an estimation of what cost might be under different conditions (e.g. using square footage from previous projects to provide a parameter that can be flexed to scale up or down for future, differently sized, projects);
- **Bottom-up Estimation** – This takes cost estimates from the lowest (bottom) level and rolls them up to produce an overall expectation of the total cost for the service, project or programme. Here, benchmark costs or historical actuals are used to provide a granular, detailed profile of costs that are then aggregated into a complete cost profile (see IPA's [Best practice in benchmarking](#) for further information on benchmarking). This estimation technique can be time consuming although can provide a higher degree of accuracy than other approaches; and
- **Three Point Estimation** – This technique uses three different estimates of potential costs, the most likely, optimistic and pessimistic profiles. These estimates are then aggregated and averaged to provide estimates that are adjusted for extremes. It is often applied in the context of Monte Carlo simulation to produce probabilistic outputs.

The choice of technique or techniques to apply will be based on a broad range of factors. These include, although are not limited to, the required fidelity of estimates (e.g. +/- 10%), access to data, and the time available. Each technique has pros and cons and a thorough understanding of these as well as how to apply the technique appropriately requires suitably qualified and experienced personnel.

At a more detailed level, when generating, manipulating and presenting estimates, decisions will have to be made. For example:

- **Basic Decision** - is use of the Mean, Mode or Average the most appropriate for a particular data set?
- **Complex Decision** - how should Monte Carlo output totals be profiled to show in-year affordability whilst taking account of probabilistic risk profiles?

**An inappropriate decision may render cost estimates unfit for purpose and suitably qualified and experienced personnel should always be engaged at the right time to both help answer and to provide Quality Assurance over these decisions.**

You should consult the Cabinet Office Sourcing Programme via [sourcing.programme@cabinetoffice.gov.uk](mailto:sourcing.programme@cabinetoffice.gov.uk) to understand where expertise may lie across government. External support may also be appropriate, including that available from specialists in these areas, for example:

- **ICEAA** – International Cost Estimating & Analysis Association (<https://www.iceaaonline.com/>)
- **ACostE** – Association of Cost Engineers (<https://www.acoste.org.uk/node/1>)
- **SCAF** – Society for Cost Analysis and Forecasting (<http://www.scaf.org.uk/>)
- **APM Estimating Guide** – Association for Project Management – ACostE Estimating Guide (<https://www.apm.org.uk/book-shop/apm-acoste-estimating-guide/>)



© 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](https://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.