



# **Forensic Science Regulator**

# **Guidance**

**Software Validation For DNA Mixture  
Interpretation**

**FSR-G-223**

**Issue 2**

© Crown Copyright 2020

The text in this document (excluding the Forensic Science Regulator’s logo, any other logo, and material quoted from other sources) may be reproduced free of charge in any format or medium providing it is reproduced accurately and not used in a misleading context. The material must be acknowledged as Crown Copyright and its title specified.

This document is not subject to the Open Government Licence.

## Contents

<b>1.</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Background .....	5
1.2	Statistical Approaches for Mixture Interpretation.....	5
1.3	Publications on DNA Mixture Interpretation Software.....	7
1.4	Validation.....	9
1.5	Software Development Methodologies.....	10
1.6	Standards for Software Development and Validation.....	10
<b>2.</b>	<b>Purpose And Scope .....</b>	<b>12</b>
<b>3.</b>	<b>Implementation.....</b>	<b>12</b>
<b>4.</b>	<b>Modification .....</b>	<b>12</b>
<b>5.</b>	<b>Terms And Definitions .....</b>	<b>13</b>
<b>6.</b>	<b>Mixture Interpretation Software Validation Requirements.....</b>	<b>14</b>
6.1	Validation Considerations Specific to Likelihood Ratio Calculations .....	14
6.2	DNA Mixture Interpretation Software Standards.....	15
<b>7.</b>	<b>Validation Process .....</b>	<b>19</b>
7.1	Overview of the Validation Process.....	19
7.2	Determination of End-User Requirements and Specification .....	21
7.3	Risk Assessment of the Process/Method.....	24
7.4	Review and Setting Acceptance Criteria .....	24
7.5	Validation of Statistical Model.....	25
7.6	Software Development and Testing.....	28
7.7	Functionality Testing .....	29
7.8	System Validation: Test of the Forensic Process .....	30

7.9	Validation Report.....	30
7.10	Validation Library .....	31
7.11	Statement or Certificate of Validation Completion .....	32
7.12	Implementation Plan .....	33
8.	End-user Validation And Validity Of The Forensic Process .....	35
8.1	End-User Validation Requirements.....	35
8.2	General Testing Requirements for End-User Validation .....	36
8.3	Minimum Testing Requirements .....	37
9.	Acknowledgements.....	38
10.	Review .....	38
11.	References .....	39
12.	Further Reading .....	432
13.	Abbreviations .....	44
14.	Glossary.....	45
Annex 1	.....	49
15.	Software Development Methodologies.....	49
Annex 2	.....	53
16.	Standards For Software Life Cycle Processes .....	53

## **1. Introduction**

### **1.1 Background**

1.1.1 Validation is the process of providing objective evidence that a method, process or device is fit for the specific purpose intended, i.e. can be relied upon. The Criminal Practice Directions [1] suggest that the court takes into account when determining the reliability of expert opinion,

“V 19A.5 (a) the extent and quality of the data on which the expert’s opinion is based, and the validity of the methods by which they were obtained.”

1.1.2 The interpretation of mixed DNA profiles including issues of subjectivity and software validation have been raised in a number of court cases, including R. v. Dlugosz and Ors [2013] EWCA, Crim 2. A closed mixtures collaborative study was commissioned by the Forensic Science Regulator in which the forensic units (FUs) in the UK participated, at a time when FUs had limited experience of probabilistic software. This study identified a high degree of consistency in the designation of the DNA profiles, but there was also a high degree of inter-laboratory and some intra-laboratory variation in the evaluating and reporting results. Some but not all participants in the study utilised DNA mixture interpretation software. One of the follow-up actions from this study, required by the Regulator, was the provision of DNA mixture interpretation software performance and validation guidance. This requirement is addressed by this document.

### **1.2 Statistical Approaches for Mixture Interpretation**

1.2.1 The general methodology for DNA mixture deconvolution that provides the basic approach for DNA mixture interpretation software tools was developed in the 1990s and has been comprehensively documented [2,3,4,5].

1.2.2 DNA mixture results generated within forensic laboratories utilising highly informative short tandem repeat (STR) multiplex amplification systems are inherently complex. Consequently, in order to deconvolute successfully and provide statistical weight to these DNA mixtures, probabilistic models for mixture interpretation have been developed. These are superseding earlier so-called binary or threshold models in which genotypes are either excluded

(probability = 0) or included (probability = 1) by considering the distribution of peak heights or areas. In contrast probabilistic models may utilise biological modelling, statistical theory, computer algorithms and probability distributions to calculate likelihood ratios (LRs) and/or infer genotypes for the DNA results. These probabilistic methods have the potential to improve the consistency and transparency of reported results. Based on the manner that peak heights are modelled, essentially there are two probabilistic methods:

- a. discrete (1.2.3); and
- b. continuous (1.2.4).

1.2.3 Discrete methods use observed peaks and incorporate a probability of allele drop-out and drop-in to explain missing or extra alleles. However, they do not take into account other variables such as peak height ratios, mixture ratios, and stutter percentages in the calculation (the caseworker uses guidelines to assess whether the stain profiles are likely to be obtained under the proposition proposed). Therefore whilst software based on this methodology has the advantage of running relatively quickly, the drawback is that not all of the available information is being used. Furthermore programs based on this approach require the operator to assign peaks as either 'stutter' or 'allelic' prior to interpretation, though a couple of programs do allow an ambiguous allele designation as 'stutter or allelic'. Some programs use the peak height information to determine the probability of allelic drop-out based upon a degradation curve determined from a dilution series.

1.2.4 Continuous methods assign a probability density for the observed profile given each possible genotype combination. This utilises the heights of all of the peaks that the analyst decides to include in the calculation. This provides a significant benefit in that it does not require the analyst to make a judgement call as to whether a given peak is allelic, stutter or over stutter. However, the user must still identify and remove any artefacts in the profile such as incomplete adenylation, spikes, or crosstalk.

1.2.5 Both discrete and continuous methods lead to the calculation of a LR. This is the measure of the weight of evidence provided by the observations (E) in relation to two propositions or hypotheses,  $H_p$  and  $H_d$ , that represent,

respectively the positions that the prosecution and defence will take at court [6]. In its simplest form, the LR is the ratio of two probabilities: the probability of the observations given that  $H_p$  is true, divided by the probability of the observations given that  $H_d$  is true:

$$LR = \frac{\Pr(E|H_p, I)}{\Pr(E|H_d, I)}$$

1.2.6 Where ‘P’ represents the probability and ‘I’ represents all of the background information that is relevant to the interpretation.

### **1.3 Publications on DNA Mixture Interpretation Software**

1.3.1 A number of scientific papers on DNA mixture interpretation software have been published that include some information on validation [7,8,9,10,11,12].

1.3.2 Only very recently have specific standards and guidance emerged pertaining to DNA mixture interpretation software validation.

- a. The Scientific Working Group on DNA Analysis Methods (SWGDM) guidelines [13] were published in (2015). These are very general in nature.
- b. Haned et al [14]. provides definitions and illustrations for the validation of probabilistic genotyping software for use in forensic DNA casework.
- c. In September 2016 the US President’s Council of Advisors on Science and Technology (PCAST) published a report on ensuring the scientific validity of feature-comparison methods [15]. PCAST emphasised that evaluation of software should not just be left to the developers. Establishing scientific validity requires scientific evaluation by other groups not involved in developing the method. Further, PCAST urged sharing within the forensic community, through publication, of high-class validation studies that properly establish the range of reliability of methods for the analysis of complex DNA mixtures.
- d. The International Society for Forensic Genetics (ISFG) has also published guidelines in 2016 for the validation of software performing bio-statistical calculations for forensic genetic calculations [16]. This stipulates the

minimum requirements for validation and covers both developmental and internal validation.

- e. The European Network of Forensic Science Institutes (ENFSI) guideline for the internal validation of software for DNA mixture interpretation [17] focuses on how the internal validation of software should be conducted on a software package that has previously been subject to full developmental validation.
- f. A recent landscape study of DNA mixture interpretation software [18] identified the potential benefits of utilising such software, which confirmed similar findings to the Regulator’s collaborative study (1.1.2). The benefits identified are included in Table 1. Availability of this type of software falls into the categories of:
  - i. freeware;
  - ii. open-source;
  - iii. commercially available products; and
  - iv. in-house solutions developed by some of the larger FUs.

1.3.3 There is a pressing need therefore to provide guidance and standards that address how validation should be approached for all types of available software.

1.3.4 The benefits of using DNA mixture interpretation software compared with manual calculations are as follows:

- a. Consistency: Reduced scope for operator-to-operator variation in data input and interpretational approach, thereby increasing consistency within and between organisations utilising the same software.
- b. Information utilisation: Software enables more sophisticated modelling that utilises the available information in the profile more efficiently. In principle, this leads to higher LRs in cases where  $H_p$  is true and smaller LRs in cases where  $H_d$  is true.
- c. Deconvolution of genotypes: This is far more effective with software, enabling database searches that would not otherwise be feasible.
- d. Improved reliability: There is a methodical approach with defined standards built on principles that have been tested and validated.



Increased automation of processing reduces the risk of human error in manual data manipulation.

- e. Reduced variability between analysts: Less analyst decision-making in terms of determining whether peaks are true alleles or artefacts, making peak assignment more automated and reducing variability between analysts.
- f. Cost-effectiveness/ utility: Increases the range of DNA profiles suitable for interpretation, including low template and complex DNA mixtures, for which manual calculation is unfeasible.
- g. Demonstrable scientific acceptance: Publication in peer-reviewed journals of the validation of the statistical models and software programs demonstrates scientific acceptance, as may be required by the courts and for compliance to BS EN ISO/IEC 17025:2017.

## **1.4 Validation**

1.4.1 General guidelines on validation are provided within Section 20 of the Forensic Science Regulator’s Codes of Practice and Conduct (the Codes) [19] plus in a separate validation guidance document to the Codes [20]. Of necessity, the aforementioned documents are generic and do not cover the validation of specific topics/techniques in any great depth. They do, however, explain the general principles that are applicable to all validation exercises, including the validation of DNA mixture interpretation software. In a very brief outline, the validation of scientific methods is defined in the Codes as:

“The process of providing evidence that a method, process or device is fit for the specific purpose intended”

and this process includes the following.

- a. Determine the end-user requirements and specification.
- b. Undertake a risk assessment of the method.
- c. Review the end-user requirements and specification.
- d. Set the acceptance criteria.
- e. Generate a validation plan.
- f. Undertake the validation exercise and record outcomes.
- g. Assess compliance with the acceptance criteria.

- h. Generate a validation report.
- i. Create a validation library.
- j. Produce a statement of validation completion.
- k. Produce an implementation plan.

1.4.2 Further generic principles that are detailed in the Codes include the following.

- a. The determination of uncertainty of measurement.
- b. Drawing a distinction between developmental validation (see also guidelines from SWGDAM [21] and ENFSI [22,23]) in which a user produces reproducible evidence for relevance, reliability and completeness themselves, and internal validation where end users are provided with a method that has already been validated by a third party. With the latter, end users seek to demonstrate that the method is fit (or remains fit) for the specific purpose intended, by providing evidence that the organisation's own competent staff can perform the method at a given location, to achieve the required outcomes.

## **1.5 Software Development Methodologies**

1.5.1 Ensuring that software is fit for purpose cannot be achieved simply by testing the software once it has been written. The software must be developed within a quality framework to ensure that the end result has been developed to the required standard, ideally through an iterative process of development, testing, and error correction.

1.5.2 There are several long-established and successful processes/sets of principles/structured programmes available for the management of the software development life cycle. These essentially comprise development, testing and implementation. In very general terms the methodologies can be described as sequential or iterative, and outline examples are given in Annex 1.

## **1.6 Standards for Software Development and Validation**

1.6.1 The ISO standards most commonly applied to forensic science undertaken within laboratories are ISO 9001 [24], and BS EN ISO/IEC 17025 [25]. The latter standard specifies general requirements for the competence of testing and calibration laboratories and is widely considered to be the most appropriate

quality standard for forensic laboratories. It needs to be used in conjunction with ILAC-G19:08/2014 [26], which translates the requirements of these standards into guidelines for forensic laboratories in order to demonstrate compliance with the standards and satisfy the specific needs of the criminal justice system.

1.6.2 The standards cited in 1.6.1 provide little by way of guidance regarding software validation. For example, BS EN ISO/IEC 17025:2017 defines validation (3.9) as; “verification (3.8), where the specified requirements are adequate for an intended use”

1.6.3 Where verification is defined as;  
“provision of objective evidence that a given item fulfils specified requirements”.

1.6.4 Software is included as an example of equipment (clause 6.4.1) that;  
“is required for the correct performance of laboratory activities and that can influence the results”

1.6.5 As such it is subject to appropriate calibration, verification and validation to demonstrate that it is fit for its intended use.

1.6.6 However, there is material available to assist with this, for example, a scheme supported by the United Kingdom Accreditation Service (UKAS) is TickITplus. This is a certification program that enables the very generic requirements of the ISO 9001 quality standard to be translated and applied to companies in the software development and computer industries. In essence this provides a practical framework for the management of software development quality through use of effective quality management system certification procedures.

1.6.7 The international standard BS ISO/IEC 12207:2008 [27] provides a common framework for software life cycle processes. This has been harmonised with another international standard, BS ISO/IEC 15288:2015 [28]. This is described in more detail in Annex 2.

1.6.8 BSI PAS 754:2014 [29] defines the overall principles and requirements for software trustworthiness with an approach that is designed to cover all the aspects of the system and software life cycle (BS ISO/IEC 15288) applicable to an organisation (the trustworthy software framework). This PAS identifies tools, techniques and processes, and addresses reliability, availability, resilience, safety and security issues.

## **2. Purpose and Scope**

- 2.1.1 The purpose of this document is to provide guidance to assist organisations in the validation of autosomal DNA mixture interpretation software and Y- STR profiling software applications. This document expands and builds upon some of the elements of the existing Forensic Science Regulator’s validation guidelines, specifically to assist in the validation of this highly specialised software application.
- 2.1.2 The scope of these validation standards and guidance encompasses all DNA mixture interpretation software programs, whether they have been purchased as a commercial package, acquired as freeware, open-source, or developed in-house. What needs to be done for each of these is summarised in Figure 2 (see 7.1.4), which compares the individual elements required in the validation of an in-house development by an end user with separate developmental and internal validation exercises. An explanation of validation requirements where a commercial package is to be used is provided in Section 8: End-User Validation and Validity of the Forensic Process.
- 2.1.3 Whilst these guidelines are intended for software programs that analyse autosomal mixtures, relevant sections also provide direct read-across to the principles that should also be applied to any DNA mixture interpretation software including the analysis of single source DNA profiles.

## **3. Implementation**

- 3.1.1 This guidance is available for incorporation into a forensic unit’s (Software provider /developer if not the forensic unit ) quality management system from the date of publication. The Regulator requires that the Codes [19] are included in the forensic units schedule of accreditation by October 2017 and the requirements in this guidance are implemented by October 2018.

## **4. Modification**

- 4.1.1 This is the **second** issue of this document.
- 4.1.2 Significant changes to the text have been highlighted in grey.

4.1.3 The modifications made to create Issue 2 of this document were to ensure compliance with The Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018. There is an updated copyright statement, some reformatting, and provision of text alternatives where information has been presented in a non-text format. Any references that have necessarily changed with the passage of time have been refreshed. The content of the document is otherwise unchanged save for in Section 4.

4.1.4 The Regulator uses an identification system for all documents. In the normal sequence of documents this identifier is of the form 'FSR-#-###' where (a) the '#' indicates a letter to describe the type or document and (b) '###' indicates a numerical, or alphanumerical, code to identify the document. For example, the Codes are FSR-C-100. Combined with the issue number this ensures each document is uniquely identified.

4.1.5 In some cases, it may be necessary to publish a modified version of a document (e.g. a version in a different language). In such cases the modified version will have an additional letter at the end of the unique identifier. The identifier thus becoming FSR-#-####.

4.1.6 In all cases the normal document, bearing the identifier FSR-#-###, is to be taken as the definitive version of the document. In the event of any discrepancy between the normal version and a modified version the text of the normal version shall prevail.

## **5. Terms and Definitions**

5.1.1 Some terms set out in FSR-G-222 DNA Mixture Interpretation [30] also apply to this document. The main technical terms employed in this appendix are listed in Section 14, the Glossary.

## **6. Mixture Interpretation Software Validation Requirements**

### **6.1 Validation Considerations Specific to Likelihood Ratio Calculations**

6.1.1 Validation of a laboratory procedure that, for example, measures a physical value can be readily undertaken by demonstrating that the measured value consistently falls within an acceptable range relative to the true value. However, with a likelihood ratio (LR) there is no ‘true’ value as such so the aforementioned validation approach applicable to metrological systems is not feasible. There are three strands to this issue.

#### **Demonstrating that the model is an acceptable approach**

6.1.2 The statistical interpretation is based on a model or series of models that seek to emulate mathematically how a biological/chemical system behaves in real life. In this specific instance it is the analysis of human DNA mixtures by extraction, amplification of short tandem repeat (STR) loci, then electrophoresis to resolve the DNA amplification polymerase chain reaction (PCR) products by size according to their electrophoretic mobility. Examples pertinent to mixture interpretation software include allelic and stutter peak height models for a continuous method of DNA interpretation [8]. The appropriateness of a model and its general acceptance by the scientific community is best demonstrated through the publication of an assessment and experimental evaluation (i.e. validation) of the model in an appropriate peer-reviewed scientific journal or through an independent review. This may be undertaken as part of the overall validation of a new software interpretation package, or more typically as a separate exercise.

#### **Demonstrating the performance of the models in cases where the true state is known**

6.1.3 In a typical case, the model will be employed to consider the observations under two propositions – representing the prosecution and defence positions respectively. A value approach to validation is to consider observations where

the true state is known ‘ground-truth’ cases. Desirable features of the model are:

- a. large LRs (greater than one) in cases where the prosecution proposition is known to be true; and
- b. small LRs (smaller than one) in cases where the defence proposition is known to be true.
- c. Ground-truth cases can be made up in various ways; by experiment, simulation and carefully selected casework data.

**Demonstrating that the calculations made by the software emulating the model are correct when the ‘true’ state is not known**

6.1.4 For other software applications used in forensic science in which output is expressed as a LR, such as paternity testing calculations, this can be demonstrated to an extent by a comparison of software outputs with expected outcomes generated by manual calculations. However, given their complexity, this poses a major challenge for mixtures interpretation models, as manual calculation is effectively impracticable for the more complex mixture calculations. A solution to this issue is to write the software programs in two different programming languages using separate programmers working independently of each other, or to repeat large blocks of calculation by hand (or using software such as Excel). Concordance of outputs, given the same input, from the two separately developed calculation methods, supplemented by manipulation of the inputs *in silico*, provide a high degree of assurance regarding the reliability of the calculations for the given statistical model [31].

**6.2 DNA Mixture Interpretation Software Standards**

6.2.1 In preparation for generating this software validation standards and guidance document, forensic units (FUs) in the UK and abroad were invited by means of a questionnaire to express their views on several aspects of standards for this type of software. Responses were used as the basis to define the following:

- a. desired performance parameters;
- b. principles that should be incorporated into a DNA mixture interpretation model; and

- c. routine operating quality checks required and data input considerations, including minimum standards for a profile to be considered suitable for interpretation.

### **Desired performance parameters**

- 6.2.2 The software should be capable of analysing three-person mixtures as a minimum.
- 6.2.3 Modelling should allow for allelic drop-in, drop-out, and ideally also stutter peaks (single and over stutter, plus single forward on an allele-specific basis, or at least locus-specific, and accounting for forward stutter tending to be at much lower frequencies than back stutter). All models should be published in peer-reviewed journals or reviewed independently.
- 6.2.4 The software should allow for easy input of data as simple text files (for example, .txt or .csv) from Microsoft Excel or analysis software (for example, GeneMapper ID-X / SoftGenetics Genemarker export files). Whilst manual entry should also be possible, it should be minimised or avoided altogether whenever possible, because of risk of transcription errors and the difficulty of recording what data have been input.
- 6.2.5 A report is desirable, for example, with data exported as text file (.txt or .csv) or as a portable document format (pdf), Microsoft Word, Microsoft Excel or SAP Crystal reports. The report should contain:
  - a. all relevant information used in the calculation, for example, databases used, Fixation Index ( $F_{ST}$ ) (co-ancestry) value; and
  - b. the alternative scenarios considered to enable checking, auditing and defence review, and the reproduction of results.
- 6.2.6 Additional information to include in the output report is pertinent information of software used: name, source, version, release date. A unique fingerprint for the report (for example, a MD5SUM) should be considered for ensuring no tampering. A non-editable report (for example, a pdf copy of the results) rather than a simple text file that could be manipulated to change the results is needed for the defence review.
- 6.2.7 The user interface should facilitate ease of use, for example:



- a. using a graphical user interface (GUI) rather than requiring command lines;
- b. allowing 'drag and drop of files'; and
- c. with the ability to navigate to profile files for analysis, select allele frequency database and other options.

6.2.8 Issues of relatedness – LR calculations are required for:

- a. unrelated;
- b. parent/child;
- c. full siblings;
- d. half-siblings;
- e. uncle/nephew; or
- f. cousins; and
- g. an expression of a unified LR, if required.

6.2.9 Population genetic issues: The ability to specify a range of ethnic databases is essential. Also the ability to select a range of Fixation Index ( $F_{ST}$ ) values. In the future it would also be desirable to account for linkage effects of syntenic loci, stratification, and a suitable method for accounting for sampling uncertainty, based on published research, for example, highest posterior density (HPD).

6.2.10 Mixture deconvolution for the purposes of database searches is considered to be highly desirable, ideally with ranking of genotype combinations.

6.2.11 Requirements for the output of calculations: These should be clear and concise but with the option of complete access where required (for example, visibility of input data, diagnostics, LR per locus, and genotype weightings) to avoid potential disclosure issues in court.

6.2.12 Sensitivity analysis: A desirable function to allow the user to consider the sensitivity of the output to a selected range of inputs.

**Principles that should be incorporated into a DNA mixture interpretation model**

6.2.13 Continuous methods take into consideration the majority of the information in the result, and are therefore considered to be the best approach scientifically.

However, discrete or binary approaches are also acceptable if fully validated. Limitations of all approaches should be made apparent to the customer.

6.2.14 Some statistical methods such as maximisation methods and Markov Chain Monte Carlo (MCMC) do not generate precisely the same number each time the same calculation is repeated. This is not a problem, as long as the variation of the numbers does not affect the number reported in court, and provided that the user is:

- a. fully trained in the use of the software;
- b. aware of the trade-off between complexity, information content, run time and precision; and
- c. able to explain those issues in layman's terms.

6.2.15 Meaningful precision should be reported by the software during validation. Overall, it is necessary to be aware that absolute precision in the evidential weight presented to a court is not necessary. In many situations, an order of magnitude for the LR is sufficient. For example, '1 billion' is suited to court use; '1.135 billion' is unlikely to be a justifiable level of precision and would in any event add no extra value in a court setting. See also Section 7.9.1c.

6.2.16 Routine operating quality checks are required and data input considerations, including minimum standards for a profile to be considered suitable for interpretation.

- a. An assessment of the stain profile (in the context of case circumstances, where possible) should always be undertaken before the use of software. The person making that assessment should be competent in the use of the software and thus aware of its limitations, in terms of the number of contributors, low template effects, etc. A calculation should proceed only if the software is considered capable of aiding a meaningful interpretation.
- b. Routine operating quality checks should be undertaken, including input data checks, settings checks, standard administrative checks, and a file review. A review of the weights by operators and the system should also be able to flag to the analyst potential issues through a set of generated diagnostics.

- c. The software used should be validated, and levels of access controlled. So, for example, only the input variables can be defined by the operator, whilst access to files that define the analytical parameters would require a higher level of authorisation. System access logs, settings changes and parameters used for past tests should be auditable.

## **7. Validation Process**

### **7.1 Overview of the Validation Process**

7.1.1 The process of validation as defined in the Codes [19] is generally applicable to the validation of all types of forensic processes and techniques. This covers both developmental validation and internal validation, which are both undertaken when a process or technique new to forensic science is developed and subsequently implemented within the same organisation. Figure 1 summarises the stages undertaken in this process, and further details are provided in Forensic Science Regulator's (FSR's) Validation, FSR-G-201 [20].

7.1.2 In the validation of mixtures interpretation software, the validation process is expanded to include three additional stages:

- a. validation of the statistical model;
- b. software development and testing; and
- c. user acceptance testing.

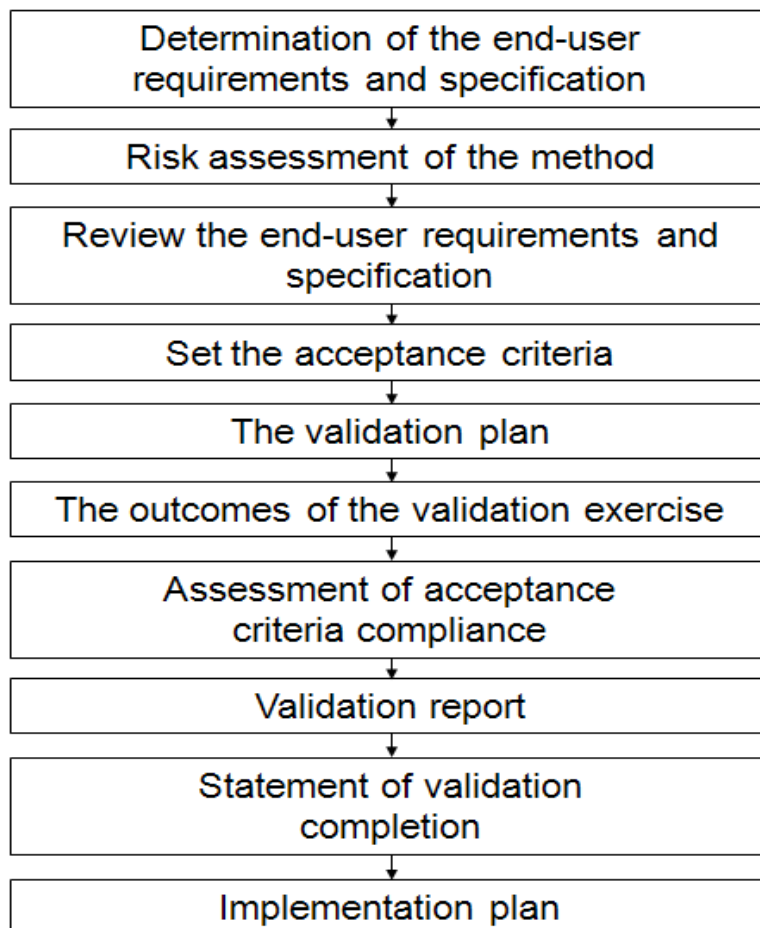
These are described in Sections 7.5 to 7.7.

7.1.3 Frequently the developmental and internal components of the overall validation process may be undertaken separately and by different organisations. For example, in the case of mixture interpretation software a commercial company may develop a software package that is sold to forensic units (FUs) for use in forensic casework applications. Under these circumstances the validation exercise is typically split.

- a. The developmental validation is undertaken by the commercial company, culminating in the release of a software package to forensic science end users.

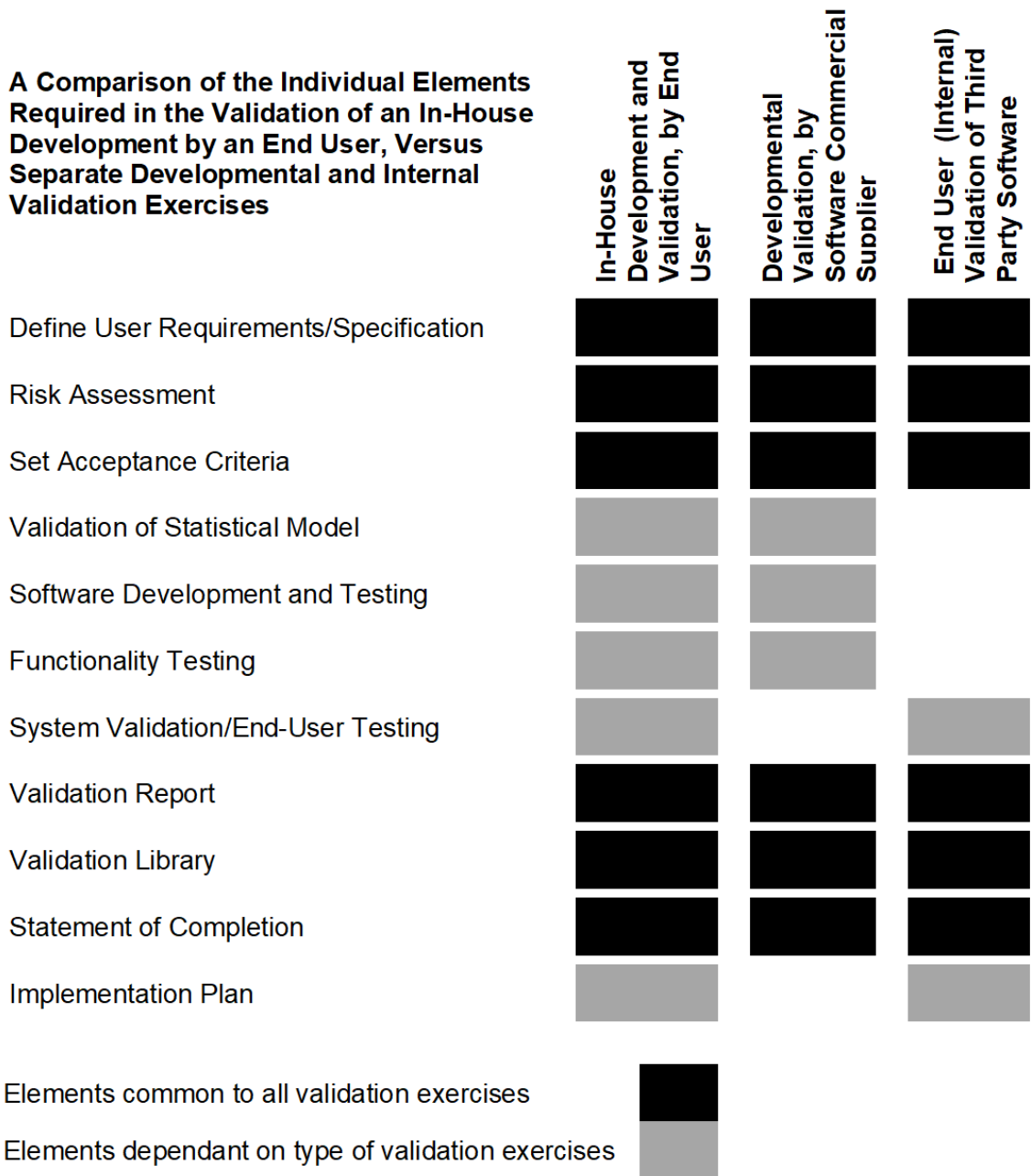
- b. The forensic science end users then conduct their own internal validation of the software to ascertain that it is fit for purpose under the conditions in which it is intended to be used.

Figure 1: Summary of the Generic Validation Process



7.1.4 The components for each of these validation exercises is defined in Figure 2, and descriptions of each component in an in-house development and validation exercise by an end user are given in Sections 7.2 to 7.12. It is recognised that many FUs will utilise commercially available software so the end-user testing that is required under these circumstances is expanded upon in Section 8.

Figure 2. Validation Processes



**7.2 Determination of End-User Requirements and Specification**

7.2.1 These have been defined in Section 6.2 from the perspective of FU end users utilising the software for interpretation and subsequent court reporting purposes. However, the impact of implementing this software is potentially profound, as it increases the range of DNA profiles suitable for interpretation, including low-template and complex DNA mixtures. Therefore, other interested parties within the criminal justice system (CJS) should also be considered here, for example,

the investigative and prosecution agencies, the defence, and reviewing authorities. All of these can also be considered to be end users.

7.2.2 Aside from the requirements they share with FUs of transparency and accuracy of reporting, these other interested parties will have additional requirements, including a clear explanation (guidance document) enabling the principles of the software to be understood, at least in outline, by non-specialists. For example, an explanation is required of why there is such a variation in the apparent weight of evidence from one approach to the next. The extent by which the range of DNA profiles that can be analysed is expanded also needs to be defined for the benefit of prosecuting authorities, which may wish to prioritise re-assessment of the DNA evidence in previously unresolved specific historic cases.

7.2.3 Typically the user requirement should be expanded into a detailed specification that defines what the software should do and how it interfaces with other systems, etc. This may include the following:

- a. data formats for inputs and outputs;
- b. how the operator interacts with the system, for example, a specification for a graphical user interface (GUI) rather than being required to type in software commands;
- c. hardware requirements – the specification of computers on which the software will be run, for example, on reporting scientists' laptops, and central storage of data;
- d. the provision of audit trail and measures to ensure data security, such as logs with timestamps, including software version and release date, operator-defined inputs such as filenames, all default parameters unchanged by operator;
- e. Software requirements, record of any dependencies, operating systems and third party libraries; and
- f. Software support availability.

Table 2: Example of a Risk Assessment Table, with Examples of Potential Errors and Suggested Control Measures

<b>Risk Description</b>	<b>Unmanaged Risk</b>	<b>Control Measures to Reduce Risk</b>	<b>Managed Risk</b>
Unsuitable profile entered	Moderate	Extensive training and competency assessment of staff	Low
Poor assessment of number of contributors to mixture	Moderate	Calculated statistic for a given DNA profile is independently corroborated by a second scientist	Low
Incorrect statistical calculation resulting in a likelihood ratio that is misleadingly large or small	Moderate	Ensure statistical model is sound through conceptual and operational validation of the model, plus peer review and publication	Low
		Back-to-back testing of the software coded in two different languages, to minimise the risk of potential coding errors	
Errors in reporting conclusions arising from analysis	Moderate	Technical check prior to closing case	Low
		Ensure that staff are appropriately trained	
		Implement regular case audit and quality assurance trials to check for the correct and consistent interpretation of analytical results	
Use of incorrect data or out-of-date software	Moderate	Check correct data are being used when drafting the validation plan	Low
		Once implemented, periodically review the validation including currency of software	

### **7.3 Risk Assessment of the Process/Method**

7.3.1 The overarching potential risk to the CJS posed by using this software is that mistakes could result in incorrect information being provided to the courts, leading to a possible miscarriage of justice. The risks of mistakes need to be identified together with control measures to mitigate these risks as part of the overall validation plan. These can be considered to fall into three main categories given below, together with an example of an assessment of risks and control measures in Table 2.

- a. Input mistakes, for example, a DNA profile is incorrectly inputted and subsequently analysed.
- b. Analytical mistakes, for example:
  - i. the model on which the software is based rests on unjustifiable assumptions;
  - ii. mistakes in software coding result in inaccuracy and unreliability of function.
- c. Mistakes in implementation of the process/method, for example, staff may not be sufficiently trained to operate the software competently or to report correctly conclusions arising from the analysis.

### **7.4 Review and Setting Acceptance Criteria**

7.4.1 Following the risk assessment the specification should be reviewed to ensure that all the identified control measures and recommendations flowing from the risk assessment are addressed. The output from this should be an agreed specification against which the validation is performed and assessed against using specific, measurable and testable or observable acceptance criteria.

Acceptance criteria could be, for example:

- a. no difference observed in numerical calculations beyond a defined limit when undertaking back-to-back testing of the software coded in two different languages;
- b. under basic stress testing when using multiple users to access the system simultaneously to create high usage levels, the system still runs at or above a defined minimum level;



- c. all cases assessed during functionality testing (black box testing) deliver the expected outcomes.

## **7.5 Validation of Statistical Model**

7.5.1 Probabilistic models implemented via software provide a means to apply the commonly accepted likelihood ratio (LR) approach to enable the interpretation of mixed DNA profiles. However, model validation is not straightforward because the notion that there is ever a ‘true value’ for the weight of evidence is a misconception. Any LR is dependent on the assumptions on which the given model is built and information that the model incorporates into the calculation. There are two elements to the validation of the statistical model that in combination ensure the model is comprehensively checked and demonstrated to be fit for purpose. These are conceptual validation and operational validation.

### **Conceptual validation**

7.5.2 Conceptual validation provides assurance that the statistical method is robust. This is ideally achieved through publication in a peer-reviewed journal, with details of the statistical model together with an evaluation of various aspects of the model’s performance. The justification of a model lies in an assessment of its performance relating to two desired characteristics:

- a. in cases where the prosecution proposition ( $H_p$ ) is genuinely true, LRs should tend to be large and increase with informativeness;
- b. in cases where the defence proposition ( $H_d$ ) is genuinely true, LRs should tend to be small and decrease with informativeness.

7.5.3 For some models consideration can also be given to:

- a. investigating conformance with Turing’s theorem; the expected value of the LR, when  $H_d$  is true;
- b. undertaking non-contributor tests, for example, consider the use of Tippett plots, particularly in relation to  $\Pr(LR > k | H_d) < 1/k$ . [9]

7.5.4 Publication should both explain and justify the model theory and underlying assumptions. It should also cover:

- a. the validity of the application of the model;

- b. limits on the application of the model/theory on the basis of the information supporting it; and
- c. any additional procedures and/or safeguards that should be implemented.

In so doing, ideally the underlying data on which conclusions are based should also be made available, for example, as supplementary material within the journal or access provided online to downloadable material including all data and a full statistical description. This enables other scientists in the field to inspect it independently and verify the results obtained in order to enable general acceptance of the model concept within the scientific community. Such transparency is essential for any software used within the CJS, for which there can be no ‘secret science’.

7.5.5 An alternative approach to publication as a means of demonstrating scientific acceptance of the conceptual validation would be for a commercial supplier to commission an independent and confidential review by an external expert. This would be provided to the organisation using the software; this may be disclosable in the event of a dispute about validity.

7.5.6 In addition to 7.5.2 and 7.5.4, a statistical specification report should be generated. The purpose of this document is to describe all models used and also choices of parameters for those models. This cannot be published in a journal because it is not novel and concise. This document should be prepared and made available for disclosure where required by the courts.

### **Operational validation of the model**

7.5.7 Operational validation of the model is the determination that;

“the model’s output behaviour has sufficient accuracy required for its intended purpose or use over the domain of the model’s intended application”[32].

7.5.8 This requires a functional computer implementation of the model, which can be tested utilising user-defined test criteria that can demonstrate whether or not outputs correlate with expectations for given inputs and the software’s intended functionality. Such testing should utilise a variety of ground-truth cases for which the composition is known, and are of varying degrees of quality and complexity that represent the full spectrum of data that may typically be

encountered in casework. This should also include some extreme examples intended to ascertain that when inputs are of sufficiently poor quality the software 'fails safe'. Testing may include the following.

- a. Check that the LR for a set of propositions for any profile is lower or equal to the inverse match probability of the profile questioned under the numerator hypothesis ( $H_p$ ).
- b. Assess the interpretation of mixtures from ground-truth cases, using variable ratios of template material and differing total amounts of DNA, and testing the model when  $H_p$  is true and when  $H_d$  is true.
- c. Determine whether the LR generated for a specific profile decreases as the information content decreases and as the ambiguity increases.
- d. Determine whether the LR reduces as any information from the evidence profile is lost and with any deviation from concordance between the suspected contributor profile and the evidence profile. For example, the effect of allelic drop-out and drop-in, degradation and inhibition.
- e. Reproducibility needs to be assessed where a statistical model such as Markov Chain Monte Carlo (MCMC) is used, which does not return precisely the same number on replicate analyses of identical data. Reproducibility needs to be tested to determine the magnitude of the variation and its impact on the reported weight of evidence.
- f. Boundary testing – test and experimentally determine the impact of increasing the number of contributors to the point at which the software ceases to provide meaningful output, for example, when a known non-contributor profile produces a high LR that indicates it is a contributor (false inclusion).
- g. Benchmarking exercises should also be undertaken:
  - i. where possible, compare the software model outputs with outputs from other software that is intended to undertake the same types of analyses on DNA mixtures;
  - ii. compare the software model outputs with manual calculations – this may be feasible only for less complex data assessments.

## **7.6 Software Development and Testing**

- 7.6.1 Of equal importance to demonstrating the validity of the statistical model is to ensure that the software developed to enable the model to be applied is both accurate and reliable in its desired application. This requires that the software developed is tested, and that errors are corrected iteratively within a quality framework to ensure that the end product performs to the required standard.
- 7.6.2 Many quality measures can be undertaken to ensure that the software is fit for purpose. None of these individually ensure the error-free performance of the end product, but in combination they help to maximise the chances of the end product ultimately meeting the user requirements [33].
- 7.6.3 The following example demonstrates one approach to verifying that the software conforms to the correct specification and an appropriate level of software trustworthiness.
- a. Based on the statistical specification document programmers working independently from each other write the code in two different languages (for example, Visual C# and R) and a number of tests are run on both versions of the computer program. Concordance of results gives assurance that the coding is an accurate reflection of the statistical specification.
  - b. Back-to-back testing of the coding in two different languages should be undertaken at both unit level (where a unit is the smallest testable part of the ‘application’) and also extended to integration testing, i.e. testing of units that interact with each other within the overall computer program that will be used in the final product. Importantly this unit and integration testing should be repeated, ideally using automated testing systems, whenever there is a program change such as a ‘bug fix’, to ensure that modifications to the code in one unit do not affect other units.
- 7.6.4 Coding is written to an appropriate standard by utilising trained and competent programmers adhering to recognised coding standards.
- 7.6.5 The quality of coding is checked by a code review.

7.6.6 A minimum of 80 per cent code coverage should be achieved during testing. Areas of the application that fall below this value should be addressed in the test summary report.

7.6.7 The use of open source software presents additional challenges with regard to software development and testing as it may not have been written specifically for the intended application. Where possible, the quality measures outlined in Section 7.6.2 should be undertaken retrospectively, including the production of a statistical specification document and the generation of a parallel program written in a different code. Similarly, when combining together two or more pre-existing and previously independently validated packages, extensive functionality and regression testing would be required. The version of the software being used should be the version that was validated, with appropriate checks made when the software is updated.

## **7.7 Functionality Testing**

7.7.1 Following the work undertaken by the software developers described in Section 7.6, independent functionality testing is required. This should be undertaken, for example, prior to the release of a commercial offering to third parties. It is also a necessary step where the software has been developed in-house by a FU and should be undertaken prior to release for end-user testing. This comprises the following.

- a. Black Box Testing: The application is tested utilising test cases designed to determine whether all the elements of the detailed specification defined in Section 7.2 have been met.
- b. Security Testing: To determine whether the system protects data and maintains functionality as intended.
- c. Integration Testing: To determine whether all individual modules deliver the required functionality when used in combination.
- d. Basic Stress Testing: To determine whether the software functionality is maintained when being used simultaneously by several users.

7.7.2 Where testing fails, the software should be revised, and the new version subject to confirmation testing to demonstrate that the defect has been corrected. This should then be followed by regression testing to verify that modifications to the

software or environment have not caused any unintended adverse knock-on effects and that the system still meets requirements.

7.7.3 On completion of all the testing, all identified test faults should have been addressed. High severity faults should be fixed and re-tested whilst lower severity faults may be accepted, but only on written acceptance by the individual authorised by the organisation as competent to assess the impact of the faults.

7.7.4 To complete this stage, a test summary report, including the accepted faults, should be generated, which is reviewed and signed off by a designated manager.

## **7.8 System Validation: Test of the Forensic Process**

7.8.1 The final stage of validation is to test that the complete process within which the software is to be used is functioning as required. This is interchangeably referred to as system validation or validation of the forensic process (see Sections 8.2. and 8.3).

## **7.9 Validation Report**

7.9.1 The validation report should include the following.

- a. Outcomes of the validation tests and assessment against the acceptance criteria.
- b. A clear definition of the conditions and limitations within which the software can be utilised.
- c. Evaluation of the assessment of uncertainty.
- d. Further guidance on uncertainty of measurement is provided in the FSR's guidelines on validation [20] and by the National Physical Laboratory [34]. BS EN ISO/IEC17025:2017 clause 7.6.3 provides reference to further information [25]. The United Kingdom Accreditation Service (UKAS) publication M 3003 (2012) [35] recognises;  
"the present state of development and application of uncertainties in testing activities is not as comprehensive as in the calibration fields".  
However, it states that the;  
"laboratory should use documented procedures for the evaluation, treatment and reporting of the uncertainty".

It is difficult to determine whether or not this requirement to assess measurement uncertainty is relevant to the consideration of DNA mixtures interpretation software, given that there is no 'measurement' process involved. The software takes a set of inputs, which will have been subject to the relevant quality procedures, and delivers a set of outputs. All of the observations and parameters will be subject to uncertainty and this is reflected in the nature of the model on which the software is based. The end product is an assessment of weight of evidence, via a LR. There is no 'correct' value for the LR. There is a question with regard to the precision with which the LR should be given and experience from across the spectrum of forensic science suggests that a 'ballpark' figure is all that is required. This view is exemplified in existing policy, agreed among all FUs, to round any calculated LR greater than  $10^9$  to 1 billion; so if a sensitivity analysis on a particular case yields LRs from  $10^{10}$  to  $10^{12}$  this is of little more than academic interest. For LRs of more modest magnitudes, there is no reason to believe that anything more precise than an order of magnitude for the evidential weight is needed. Furthermore, whereas the scientist would be expected to consider issues of sensitivity there is no requirement that he/she should provide a range of LRs to the court.

## **7.10 Validation Library**

7.10.1 The provider should create and have available a 'library' of documents relevant to the validation of the software. The library should include, but need not be limited to, the following.

- a. The specification for the approved software and the process within which it is applied.
- b. The risk assessment for the approved software and the process within which it is applied.
- c. Any associated supporting material, such as academic papers or technical reports that were used to support or provide evidence on the applicability of the method.
- d. The validation plan for the approved software and the process, including user acceptance criteria.

- e. Information supporting the statistical model. Evidence of acceptance within the scientific community should be provided, such as an article published in a peer-reviewed journal, or failing that an assessment report from an independent expert.
- f. A statistical specifications report. Ideally this should include the underlying data on which any conclusions are based, and a full statistical description of sufficient detail such that other scientists in the field could independently inspect it and verify the results obtained.
- g. The validation report. This should include summaries of the data and an assessment against the acceptance criteria. The information provided must properly reflect the results obtained and be sufficient to support any conclusions drawn in the report.
- h. The record of approval and statement of validation completion.

7.10.2 Where the validation relies on material published by others, the provider should keep a copy as part of the library to ensure that the information is readily accessible. This is especially important if the source of the material is not permanent, for example, published on the internet; an 'instance' of the material should be captured and the time and date of capture recorded.

7.10.3 The validation library should be maintained by the provider to cover the period for which the method is in use; and any legal challenges to the output from the method that may arise.

## **7.11 Statement or Certificate of Validation Completion**

7.11.1 On the satisfactory completion of the validation exercise and independent assessment of the validation report, written management sign-off should be provided by means of a short signed and dated statement that summarises the validation and key issues regarding the methodology. This should include the following.

- a. A statement that the method has been approved, by whom and the nature of the approval, and the scope of the validation performed.
- b. Key applications of the method and any limitations to its use, including circumstances where the method should not be used.
- c. Signature, date and role of manager signing off the validation completion.



7.11.2 During this independent assessment the manager responsible for validation should ensure that the following have been satisfactorily addressed.

- a. The validation exercise has demonstrated that the technique is fit for purpose.
- b. Any risks to the CJS have been appropriately addressed, for example, there are no known legal issues that could potentially undermine the use of the method.

## **7.12 Implementation Plan**

7.12.1 The introduction of any new technique following validation needs to be carefully considered and planned for, to ensure that the implementation is controlled and that application of the technique continues to be fit for purpose once introduced. This plan includes a consideration of the following.

- a. Training and competency assessment. Plans for training staff in the use of the software plus the means of assessing their competency in the use of the software and interpretation of the outputs need to be in place prior to implementation.
- b. Roll-out of hardware and software. The approach being adopted to rolling out the use of the software needs to be decided upon and included in the implementation plan. For example, it might be considered advisable to conduct a pilot exercise first. This would use the analysis software on a restricted number of live cases, assessed within a single unit and using the most experienced practitioners. On a satisfactory review of the pilot, the application could then be widened to other practitioners and cases.
- c. Ensuring software version control, locking and labelling, with alignment to validation documents. A system should be in place to ensure that changes to the computer programs are appropriately managed. Changes or revisions are usually identified by a number or letter code, i.e. the 'revision number'. A control system should be in place to ensure that where multiple operators are utilising the same software, by default only the latest approved version is available to end users, for example, by controlling access through a centralised single authoritative data store.

- d. Updates to software. Whenever there is a change to the software such as an update to its functionality or a 'bug fix', aside from testing the functionality of the software unit within which the change has been made, integration testing should also be undertaken to ensure that modifications to the code in one unit have not affected other interrelated units. All details of software changes made, tests undertaken (verification), and changes to version number shall be documented and retained within the validation library.
- e. Ensuring control of information used by the software. A system should be in place to ensure that information used by the software such as allele proportions, analytical parameters, and other databases is controlled and changes to the information are appropriately managed. Changes or revisions should be identified by a 'revision number'. By default only the latest approved version is available to end users via use of the software, for example, by controlling access through a centralised single authoritative data store.
- f. Quality assurance, audit and accreditation. Use of the software needs to be captured within the laboratory regular quality assurance trials and audit programmes. Arrangements should also be made for the software validation to be independently assessed by UKAS and included as an extension to scope for accreditation to the standard BS EN ISO/IEC 17025 [25].
- g. A short (about two pages) 'Question and Answer' style document should be generated that outlines the strengths and weaknesses of the mixtures interpretation software (see the Codes clause 20.2.57 A statement of validation completion) [19]. This should be made available to the courts in the event that admissibility of the technique is questioned. It is a Crown Prosecution Service (CPS) requirement that a document of this nature be generated for scientific techniques under consideration by the courts [35].

## **8. End-User Validation And Validity Of The Forensic Process**

### **8.1 End-User Validation Requirements**

- 8.1.1 Ultimately it is the court reporting officer who is the end user of the software. This officer needs to be satisfied, through the provision of full validation documentation plus formal assessment and authorisation by their organisation, that the software they are relying upon to provide expert opinion is fit for purpose and will not result in misdirection of the court.
- 8.1.2 Validation can be considered to comprise first and foremost developmental validation as described in Section 7.1.3 plus internal validation, also known as end-user validation or verification. The latter is required where the developmental validation work has been undertaken by a different organisation or a different part of the same organisation. In this instance there is still a requirement for evidence that the end user, or the forensic unit (FU), can correctly perform the method at a given location, i.e. with their own staff using their own equipment, standard operating procedures (SOPs) and facilities. In other words, this is ‘demonstrating that it works in your hands’, or demonstrating technical competence in providing valid and accurate data and results, which is the fundamental aim of accreditation to BS EN ISO17025:2017 [25].
- 8.1.3 For each software update where the change is determined to not affect the original validation then a full validation is not required, but an appropriate verification is; this could include back-to-back comparison with the previous version.
- 8.1.4 The procedure followed for internal validation is essentially the same as that described in Section 7. However the requirements are less onerous in that this builds on work already undertaken to validate the statistical model, plus conceptual and operational validation have already been undertaken, together with testing of the software functionality by a third party. The proviso here is that the existing evidence that has been produced by a third party, and on which reliance is placed, must be relevant, available and adequate. Provided these requirements are met, end-user validation may be sufficient to meet validation

requirements, for example, where commercially available software or freeware is to be used.

### **Reliability of external evidence**

- 8.1.5 An ideal situation when wishing to utilise third-party evidence of validation, is that the required relevant, complete and objective evidence is provided as a validation study published as a peer-reviewed article in a respected scientific journal, and the details of the analysis undertaken are both transparent and accessible to third parties. Whilst not foolproof, provided the points raised in Section 7 have been fully addressed then this can go a long way to providing assurance and transparency that the science undertaken is fit for purpose.
- 8.1.6 Unfortunately validation studies are sometimes considered insufficiently novel to merit publication in peer-reviewed journals. However, other sources of information may suffice such as validation studies on the website of manufacturers/suppliers that have a well-established reputation for meeting the quality requirements of the forensic community. Published work deemed relevant and reliable may be used to supplement evidence of meeting the performance criteria derived from the user requirement/specification that need to be verified to demonstrate that the method works in the FU.
- 8.1.7 An alternative approach to publication as a means of demonstrating scientific acceptance of the conceptual validation would be for a commercial supplier to commission an independent and confidential review by an external expert. This would be provided to the organisation using the software; this may be disclosable in the event of a dispute about validity.

## **8.2 General Testing Requirements for End-User Validation**

- 8.2.1 End-user validation should include an assessment of ground-truth cases, i.e. mixtures of variable quality and complexity, assembled from known and previously typed sources in addition to testing pristine biological samples, or virtual samples. A key output from the end-user validation should be to define the parameters within which the software operates reliably, i.e. the number of contributors, ratio of minor contributors and total amount of DNA.

8.2.2 Analysis of casework material in validation exercises can provide a more demanding test than using mock casework because it is difficult to generate a set of test data that adequately reflects the diversity, quality and complexity encountered in casework. However, even the use of test material from closed cases is not without potential consequences. Mixture interpretation software offers the potential to provide evidence from samples that previously could not be assessed through manual analytical processes, and may provide much more powerful evidence in support of one of the two propositions. Therefore, it would be valuable if back-to-back comparison against the existing laboratory processes is introduced by means of a pilot on live casework once the validation exercise has been completed. This should be part of the implementation rather than an element of end-user validation. The benefit of this approach compared with using casework that has already been judicially discharged is that all material is available for further review and consideration in the event of a query being raised from the data interpretation. Additional guidance on the use of casework material for validation purposes has been published by the Forensic Science Regulator (FSR) [36]. Benchmarking by using external proficiency trials is also encouraged.

### **8.3 Minimum Testing Requirements**

#### **Determination of laboratory-specific parameters**

8.3.1 Establish the minimum quality criteria for profiles to be submitted for analysis with the software.

8.3.2 Assess all operator-adjustable default settings to ensure that these are appropriate, for example, the selection of a specific drop-in model, analytical threshold, and saturation limit for the capillary electrophoresis (CE) instrument from which the data are being submitted.

#### **Software functionality checks**

8.3.3 Check the accuracy and correct functioning of the conversion of output files from the CE instrument into input files for the interpretation software.

8.3.4 Assess genotype weightings.

8.3.5 Assess the effects of allelic drop-in and DNA amplification polymerase chain reaction (PCR) artefacts on likelihood ratios (LRs).

8.3.6 Assess the variability in LR<sub>s</sub> generated by repeat analyses.

**Evaluation of likelihood ratio behaviour to understand the range that may be expected**

8.3.7 Assess contributor LR<sub>s</sub> under high and low template conditions, with different numbers of contributors and as profile information is lost (partial profiles, inhibition).

8.3.8 Assess non-contributor LR<sub>s</sub> under a variety of conditions to consider the probability of providing misleading evidence.

8.3.9 Where the software has the required functionality, assess the effects on LR<sub>s</sub> of combining replicate analyses, and of adding conditioning profiles.

8.3.10 Where the software requires the number of contributors to be determined by the operator, use ground-truth data to assess the effect of assuming both more than and fewer than the correct number of contributors.

8.3.11 Establish a policy for allowing for sampling effects and suitable values for the Fixation Index ( $F_{ST}$ ).

## **9. Acknowledgements**

9.1.1 This guidance was produced following the award of a competitive tender to Principal Forensic Services. The authors would like to thank Cellmark Forensic Services, Professor Mike Coble (University of North Texas Health Science Center), Eurofins Forensic Services, Professor Peter Gill (University of Oslo Hospital), Forensic Science Ireland, Key Forensic Services Ltd, Scottish Police Authority, members of the Forensic Science Regulator's DNA Analysis Specialist Group and the forensic science regulation unit (FSRU).

## **10. Review**

10.1.1 This published guidance will form part of the review cycle as determined by the Forensic Science Regulator.

10.1.2 The Forensic Science Regulator welcomes comments. Please send them to the address as set out on the following web page:  
[www.gov.uk/government/organisations/forensic-science-regulator](http://www.gov.uk/government/organisations/forensic-science-regulator), or send them to the following email address: [FSREnquiries@homeoffice.gsi.gov.uk](mailto:FSREnquiries@homeoffice.gsi.gov.uk)

## 11. References

- [1] Criminal Procedure Rules and Practice Directions. Available at:  
<http://www.justice.gov.uk/courts/procedure-rules/criminal/rulesmenu-2015>  
[Accessed 23/07/2020].
- [2] Clayton, T. M., Whitaker, J. P., Sparkes, R. and Gill, P. (1998) 'Analysis and interpretation of mixed forensic stains using DNA STR profiling', *Forensic Science International*, 91, pp 55–70.
- [3] Evett, I. W., Buffery, C., Willott, G. and Stoney, D. (1991) 'A guide to interpreting single locus profiles of DNA mixtures in forensic cases', *Journal of the Forensic Science Society*, 31, pp 41–47.
- [4] Evett, I. W., Gill, P. D. and Lambert, J. A. (1998) 'Taking account of peak areas when interpreting mixed DNA profiles', *Journal of Forensic Sciences*, 43 (1), pp 62–69.
- [5] Gill, P., Brenner, C. H., Buckleton, J. S., Carracedo, A., Krawczak, M., Mayr, W. R., Morling, N., Prinz, M., Schneider, P. M. and Weir, B. S. (2006) 'DNA commission of the International Society of Forensic Genetics: Recommendations on the interpretation of mixtures', *Forensic Science International*, 160 (2–3), pp 90–101.
- [6] Puch-Solis, R., Roberts, P., Pope, S. and Aitken, C. (2012) *Practitioner Guide No 2: Assessing the Probative Value of DNA Evidence, Guidance for Judges, Lawyers, Forensic Scientists & Expert Witnesses*.
- [7] Bright, J.-A., Evett, I. A., Taylor, D., Curran, J. M. and Buckleton, J. (2015) 'A series of recommended tests when validating probabilistic DNA profile interpretation software', *Forensic Science International: Genetics*, 14, pp 125–131.

- [8] Bright, J.-A., Taylor, D., Curran, J. M. and Buckleton, J. (2013) 'Developing allelic and stutter peak height models for a continuous method of DNA interpretation', *Forensic Science International: Genetics*, 7 (2), pp 296–304.
- [9] Taylor, D., Buckleton, J. and Evett, I. (2015) 'Testing likelihood ratios produced from complex DNA profiles', *Forensic Science International: Genetics*, 16, pp 165–171.
- [10] Perlin, M. W., Belrose, J. L. and Duceman, B. W. (2013) 'New York State TrueAllele® casework validation study', *Journal of Forensic Sciences*, 58 (6), pp 1458–1466.
- [11] Perlin, M. W., Legler, M. M., Spencer, C. E., Smith, J. L., Allan, W. P., Belrose, J. L. and Duceman, B. W. (2011) 'Validating TrueAllele® DNA mixture interpretation', *Journal of Forensic Sciences*, 56 (6), pp 1430–1447.
- [12] Perlin, M. W., Dormer, K., Hornyak, J., Schiermeier-Wood, L. and Greenspoon, S. (2014) 'TrueAllele® casework on Virginia DNA mixture evidence: Computer and manual interpretation in 72 reported criminal cases', *PloS ONE*, 9 (3), e92837.
- [13] SWGDAM, (2015) Guidelines for the validation of probabilistic genotyping systems, Scientific Working Group on DNA Analysis Methods. Available at: [www.swgdam.org/publications](http://www.swgdam.org/publications) [Accessed 23/07/2020].
- [14] Haned, H., Gill, P., Lohmueller, K., Inman, K. and Rudin, N. (2016) 'Validation of probabilistic genotyping software for use in forensic DNA casework: Definitions and illustrations', *Science and Justice*, S6, pp 104–108.
- [15] PCAST (2016) Forensic science in criminal courts: ensuring scientific validity of feature-comparison methods, US President's Council of Advisors on Science and Technology. Available at: [obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/PCAST/pcast\\_forensic\\_science\\_report\\_final.pdf](http://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/PCAST/pcast_forensic_science_report_final.pdf) [Accessed 23/07/2020].
- [16] Coble, M. D., Buckleton, J., Butler, J. M., Egeland, T., Fimmers, R., Gill, P., Gusmao, L., Guttman, B., Krawczak, M., Morling, N., Parson, W., Pinto, N., Schneider, P. M., Sherry, S. T., Willuweit, S. and Prinz, M. (2016) 'DNA Commission of the International Society for Forensic Genetics:



Recommendations on the validation of software programs performing biostatistical calculations for forensic genetics applications’, Forensic Science International: Genetics, 25, pp 191–197.

[17] ENFSI (2017) Best practice manual for the internal validation of probabilistic software to undertake DNA mixture interpretation. Available at: <http://enfsi.eu/wp-content/uploads/2017/09/Best-Practice-Manual-for-the-internal-validation-of-probabilistic-software-to-undertake-DNA-mixture-interpretation-v1.docx.pdf> [Accessed 23/07/2020].

[18] Ropero-Miller, J. (2015) National Institute of Justice, Forensic Technology Centre of Excellence Landscape study of DNA mixture interpretation software, 6. Available at: <https://nij.ojp.gov/library/publications/list> [Accessed 23/07/2020].

[19] Forensic Science Regulator, Codes of Practice and Conduct for Forensic Science Providers and Practitioners in the Criminal Justice System. Birmingham: Forensic Science Regulator. Available at: [www.gov.uk/government/collections/forensic-science-providers-codes-of-practice-and-conduct](http://www.gov.uk/government/collections/forensic-science-providers-codes-of-practice-and-conduct) [Accessed 23/07/2020].

[20] Forensic Science Regulator, Validation, FSR-G-201. Birmingham: Forensic Science Regulator. Available at: <https://www.gov.uk/government/publications/forensic-science-providers-validation> [Accessed 23/07/2020].

[21] SWGDAM, (2012) Validation guidelines for DNA analysis methods, Scientific Working Group on DNA Analysis Methods. Available at: [www.swgdam.org/publications](http://www.swgdam.org/publications) [Accessed 23/07/2020].

[22] ENFSI, (2010) Recommended minimum criteria for the validation of various aspects of the DNA profiling process, Issue No. 1, European Network of Forensic Science Institutes. Available at: [http://enfsi.eu/wp-content/uploads/2016/09/minimum\\_validation\\_guidelines\\_in\\_dna\\_profiling\\_-\\_v2010\\_0.pdf](http://enfsi.eu/wp-content/uploads/2016/09/minimum_validation_guidelines_in_dna_profiling_-_v2010_0.pdf) [Accessed 23/07/2020].

[23] SWGDAM (2015) Guidelines for the validation of probabilistic genotyping systems, Scientific Working Group on DNA Analysis Methods. Available at: [www.swgdam.org/publications](http://www.swgdam.org/publications) [Accessed 23/07/2020].

[24] International Organization for Standardization, ISO 9001:2015: Quality management systems –Requirements. Available at: [www.iso.org/iso-9001-quality-management.html](http://www.iso.org/iso-9001-quality-management.html) [Accessed 23/07/2020].

[25] British Standards, BS EN ISO/IEC 17025:2017 General requirements for the competence of testing and calibration laboratories. Available at: [shop.bsigroup.com](http://shop.bsigroup.com) [Accessed 23/07/2020].

[26] International Laboratory Accreditation Cooperation, (2014) ILAC-G19:08/2014 Modules in a Forensic Science Process, International Laboratory Accreditation Cooperation. Available at: [ilac.org/publications-and-resources/ilac-guidance-series/](http://ilac.org/publications-and-resources/ilac-guidance-series/) [Accessed 23/07/2020].

[27] British Standards, BS ISO/IEC 12207:2008(E) Systems and software engineering –Software life cycle processes. Available at: [shop.bsigroup.com](http://shop.bsigroup.com) [Accessed 23/07/2020].

[28] British Standards, BS ISO/IEC 15288:2015 Systems and software engineering. System life cycle processes. Available at: [shop.bsigroup.com](http://shop.bsigroup.com) [Accessed 23/07/2020].

[29] British Standards, PAS 754:2014 Software trustworthiness – Governance and management – Specification. Available at: [shop.bsigroup.com](http://shop.bsigroup.com) [Accessed 23/07/2020].

[30] Forensic Science Regulator, DNA mixture interpretation, FSR-G-222. Birmingham: Forensic Science Regulator. Available at: [www.gov.uk/government/collections/dna-guidance](http://www.gov.uk/government/collections/dna-guidance) [Accessed 23/07/2020]

[31] Bleka, Ø., Storvik, G. and Gill, P. (2016) 'EuroForMix: An open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts', Forensic Science International: Genetic, 21, pp 35–44.

[32] Sargent, R. G. (2013) 'Verification and validation of simulation models', Journal of Simulation, 7, pp 12–24.

[33] NASA, (1992) Recommended Approach to Software Development, Revision 3, National Aeronautics and Space Administration. Available at:

[ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930009672.pdf](https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930009672.pdf) [Accessed 23/07/2020].

[34] National Physical Laboratory, (2001) Good Practice Guide No. 11: A Beginner's Guide to Uncertainty of Measurement by the National Physical Laboratory. Available at: [www.npl.co.uk/publications/a-beginners-guide-to-uncertainty-in-measurement](http://www.npl.co.uk/publications/a-beginners-guide-to-uncertainty-in-measurement) [Accessed 23/07/2020].

[35] Crown Prosecution Service, (2012) Core Foundation Principles for Forensic Science Providers. Available at: [Core Foundation Principles for Forensic Science Providers | The Crown Prosecution Service](https://www.cps.gov.uk/core-foundation-principles-for-forensic-science-providers) [Accessed 23/07/2020].

[36] Forensic Science Regulator, Validation – Use of Casework Material, FSR-P-300. Birmingham: Forensic Science Regulator. Available at: [www.gov.uk/government/publications/protocol-using-casework-material-for-validation-purposes](http://www.gov.uk/government/publications/protocol-using-casework-material-for-validation-purposes) [Accessed 23/07/2020].

## 12. Further Reading

British Standards, BS EN ISO/IEC 17020:2012 General criteria for the operation of various types of bodies performing inspection.

Gill, P., Kirkham, A. and Curran, J. (2007) 'LoComatioN: a software tool for the analysis of low copy number DNA profiles', Forensic Science International, 166 (2–3), pp 128–138.

Trimble, J. and Webster, C. (2012) Agile Development Method for Space Operations. Available at: [ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120013429\\_2012013093.pdf](https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120013429_2012013093.pdf) [Accessed 23/07/2020]

UKAS®, (2012) The Expression of Uncertainty and Confidence in Measurement, M 3003, 4th edition, United Kingdom Accreditation Service. Available at: <https://www.ukas.com/download/publications/publications-relating-to-laboratory-accreditation/M3003-Expression-of-Uncertainty-and-Confidence-in-Measurement-Edition-4-October-2019.pdf> [Accessed 23/07/2020].

### 13. Abbreviations and Acronyms

<b>Abbreviation or acronym</b>	<b>Meaning</b>
BS	British Standard
CE	Capillary electrophoresis
CJS	Criminal justice system
DNA	Deoxyribonucleic acid
EN	European norm
ENFSI	European Network of Forensic Science Institutes
FU	Forensic unit
FSR	Forensic Science Regulator
F <sub>ST</sub>	Fixation Index
GUI	Graphical user interface
HPD	Highest posterior density
IEC	International Electrotechnical Commission
ISFG	International Society for Forensic Genetics
ISO	International Organization for Standardization
LR	Likelihood ratio
MCMC	Markov Chain Monte Carlo
PCAST	President's Council of Advisors on Science and Technology
PCR	Polymerase chain reaction
PDF	Portable Document Format

RAD	Rapid application development
SOP	Standard operating procedure
STR	Short tandem repeat
SWGDM	Scientific Working Group on DNA Analysis Methods
UAT	User acceptance testing
UK	United Kingdom
UKAS	United Kingdom Accreditation Service

## 14. Glossary

### **Application Testing:**

See End-User Testing.

### **Black Box Testing:**

A method of software testing that examines the functionality of an application without considering its internal structures and workings.

### **Co-ancestry Coefficient:**

See Fixation Index ( $F_{ST}$ ).

### **Conceptual Validation:**

The provision of assurance that the statistical model is fundamentally robust.

### **Developmental Validation:**

The acquisition of test data and the determination of the conditions and limitations of a new or novel methodology for use in forensic science.

### **End-User Testing:**

This is also sometimes called application testing and user acceptance testing (UAT). It is a phase of software development in which the software is tested in the 'real world' by the intended users.

**Fixation Index (FST):**

Also referred to as ‘co-ancestry coefficient’ and is a measure of population substructure.

**Freeware:**

Software that is available free of charge.

**Ground Truth:**

A data set made from known source material, such as DNA extracted and analysed from stains produced using body fluids from known donors, used for validation, proficiency and competency testing purposes.

**Implementation:**

This has two definitions.

- a. ‘Implementation’ in the context of development of software is essentially writing the software application.
- b. ‘Implementation’ in terms of an overall programme of introducing a new process or technique or service is the process of putting the new system in place once validation has been completed. This would include:
  - i. training and competence assessment of operators;
  - ii. the physical set-up of the equipment and accommodation required at implementation;
  - iii. ensuring that the new process is included in quality assurance programmes; and
  - iv. establishing servicing and maintenance.

**Internal Validation:**

This is also often referred to as end-user validation. It is an accumulation of test data to demonstrate that an established method or procedure performs as expected in the laboratory into which it is being introduced. Prior to using a procedure for forensic applications, a laboratory shall conduct internal validation studies.

**Life Cycle:**

The evolution of a system, product, service or other human-made entity from conception through development, testing, implementation and ultimately retirement.

**Likelihood Ratio (LR):**

The ratio of probabilities (Pr) for the evidence (E) given the prosecution hypothesis ( $H_p$ ) and the defence hypothesis ( $H_d$ ) and taking account of relevant background information,

$$LR = \frac{\Pr(E|H_p, I)}{\Pr(E|H_d, I)}$$

**Markov Chain Monte Carlo (MCMC):**

Generic statistical method that, in the context of DNA mixture interpretation, uses peak heights and through a very large number of iterative computations calculates weights of all possible genotype combinations according to how well they explain the observed data.

**Mixture:**

Sample consisting of DNA from more than one individual.

**Open-Source:**

Software for which the original source code is made freely available and may be redistributed and modified.

**Operational Validation:**

The provision of assurance that the model's output behaviour has sufficient accuracy required for the model's intended purpose over the domain of the model's intended applicability.

**Software Problem Resolution Process:**

Process by which all discovered problems are identified, analysed, managed and controlled to resolution.

**Software Unit:**

The smallest testable part of an application.

**Statistical Model:**

A set of assumptions concerning the generation of the observed data, and similar data from a larger population. The model represents, often in considerably idealised form, the data-generating process.

**Syntenic Loci:**

Loci physically co-localised on the same chromosome.

**User Acceptance Testing (UAT):**

See End-User Testing

**Validation:**

The process of providing objective evidence that a method, process or device is fit for the specific purpose intended.

**Verification:**

The confirmation through the assessment of existing evidence or through experiment that a method, process or device is fit (or remains fit) for the specific intended purpose. This includes an overriding requirement that there is evidence that the forensic unit's own competent staff can perform the method at a given location.



## **Annex 1**

### **15. Software development methodologies**

15.1.1 Common methodologies include:

- a. waterfall (and V model);
- b. prototyping;
- c. iterative and incremental development;
- d. spiral development;
- e. extreme programming; and
- f. various types of agile methodology. Additional background information on ensuring that software is fit for purpose in an error-intolerant application using agile development method is provided by Trimble and Webster (2012) Agile Development Method for Space Operations.

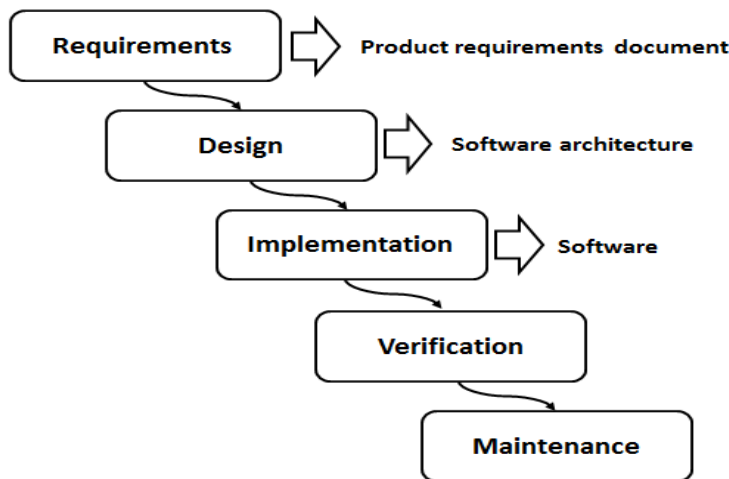
15.1.2 Many of these methodologies can be conveniently applied through the use of commercially available packages, which often include aids to streamline the whole process, for example, through use of automated testing tools. An example is the Rational Unified Process, which is an iterative software development process framework created by Rational Software Corporation, a division of IBM. The commercial product to enable the application of this framework includes a hyperlinked knowledge base with sample artefacts and detailed descriptions for many different types of activities.

15.1.3 The waterfall model is a sequential development approach in which development passes through several ordered phases, typically:

- a. generation of a software requirements specification;
- b. design of software;
- c. coding ('implementation');
- d. testing;
- e. integration (if there are multiple subsystems); and
- f. deployment (or installation).

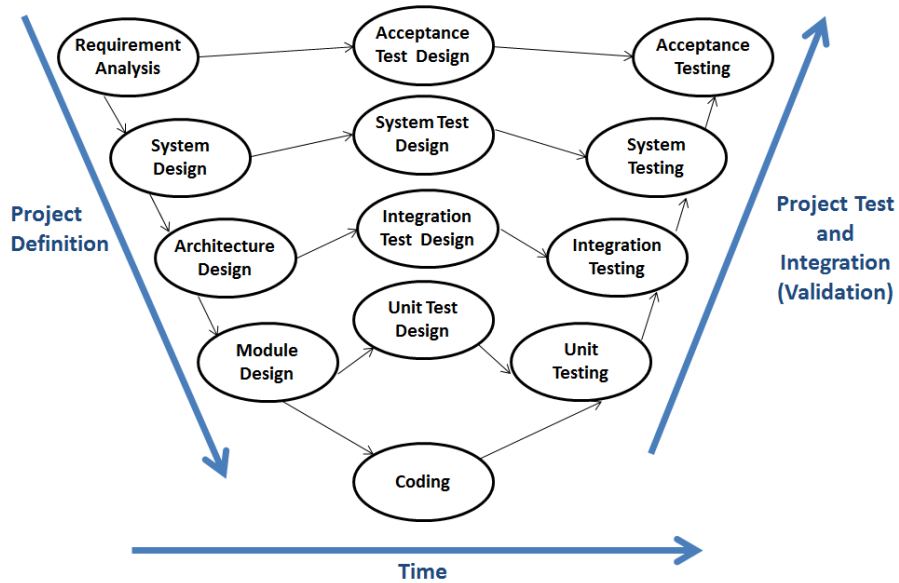
- 15.1.4 Whilst iteration can be introduced into the waterfall development process by including a prototyping stage, in practice this is often omitted resulting in a largely linear process, typically lacking in flexibility.

Figure 3: The Activities of the Software Developmental Process Represented in the Waterfall Model



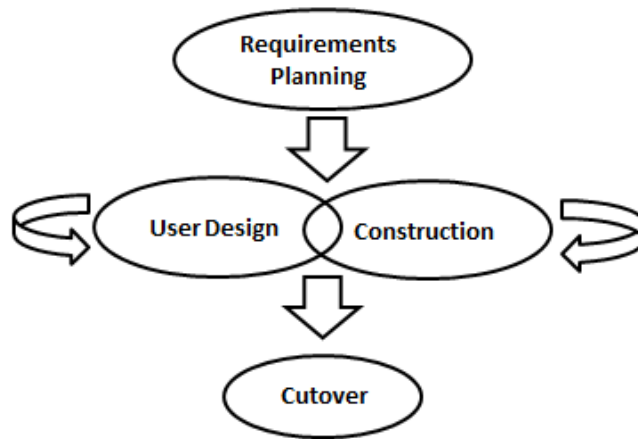
- 15.1.5 A useful progression of the waterfall development process is the so-called V model shown in the schematic diagram below. This expands on and helps to explain the verification part of the overall process by representing the requirements to be met on the left and the steps to be taken on the right. It shows that the detailed design may be broken down into functional subunits that are individually tested, followed by combining and testing these units as subsystems before finally progressing to whole system validation.

Figure 4: Diagrammatic Representation of the V Model for Verification and Validation



15.1.6 Rapid application development (RAD) is an example of a software development methodology intended to favour iterative development and rapid prototype construction, and as such is in direct contrast with sequential development approaches. The basic principles of RAD start with the development of a preliminary data model using structured techniques. The requirements are then verified using iterative prototyping to refine the data model. In recent years the rapid/iterative software development methodologies have become more popular, not least because of their greater flexibility. However, the choice of methodology framework depends on a number of factors including technical, organisational, project and team considerations.

Figure 5: Schematic Representation of the Rapid Application Development Model

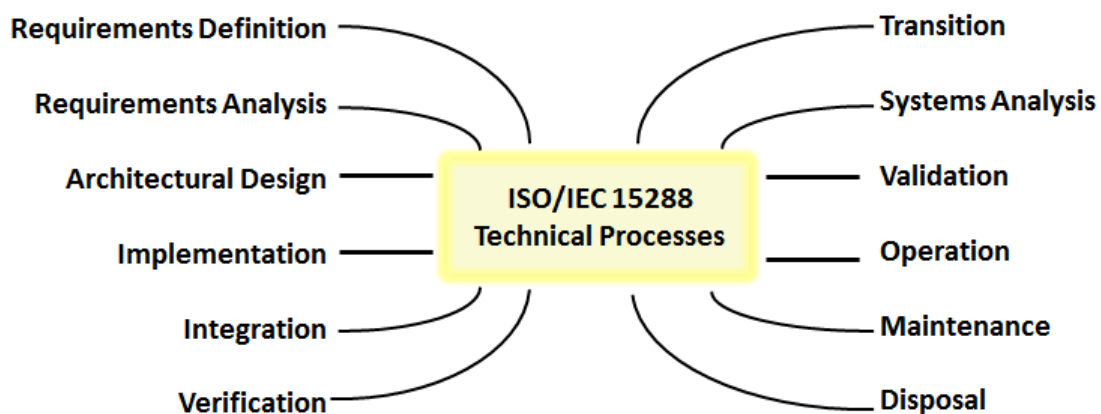


## Annex 2

### 16. Standards For Software Life Cycle Processes

16.1.1 International standard ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes provides a common framework for software life cycle processes including the development, operation, maintenance and disposal of software products, and also includes requirements for software validation and verification. In 2008 this was harmonised with another international standard, ISO/IEC 15288:2008 Systems and software engineering – System life cycle processes, and the latter has been further revised by ISO/IEC/IEEE 15288:2015. As defined in the document titles, ISO 15288:2015 establishes a common framework for describing the life cycle of all systems created by humans, whilst ISO/IEC 12207:2008 is specific to software life cycles only. It provides a comprehensive set of requirements that are applicable to all the development stages of any software, regardless of the model chosen for the development process. These are summarised in the diagram below including both software validation and verification.

Figure 6: Components of ISO/IEC 15288:2015 Standard



16.1.2 ISO/IEC 12207:2008 defines the purpose of software verification to be the confirmation that each software work product and/or service of a project or project properly reflects the specified requirements. This is broken down into specific areas namely:

- a. requirements verification;
- b. design verification;
- c. code verification;
- d. integration verification; and
- e. documentation verification.

16.1.3 ISO/IEC 12207:2008 defines the purpose of software validation to be the confirmation that the requirements for a specific intended use of the software work product are fulfilled. This is achieved through the following steps for testing (although other means may also be permissible such as modelling, analysis and simulation).

- a. Prepare a validation plan including test requirements, test cases and test specifications for analysing test results, and ensure that these reflect the particular requirements for the specific intended use.
- b. Included in the test requirements should be stress, boundary and singular input testing.
- c. Implement the validation plan. All problems and non-conformances are entered into a software problem resolution process to ensure that all discovered problems are identified, analysed, managed and controlled to resolution.

Published by:

The Forensic Science Regulator

5 St Philip's Place

Colmore Row

Birmingham

B3 2PW

<https://www.gov.uk/government/organisations/forensic-science-regulator>