iseb
Qualifications from BCS

Enabling the information society

Intermediate Certificate in Software Testing Syllabus

Version 1.4

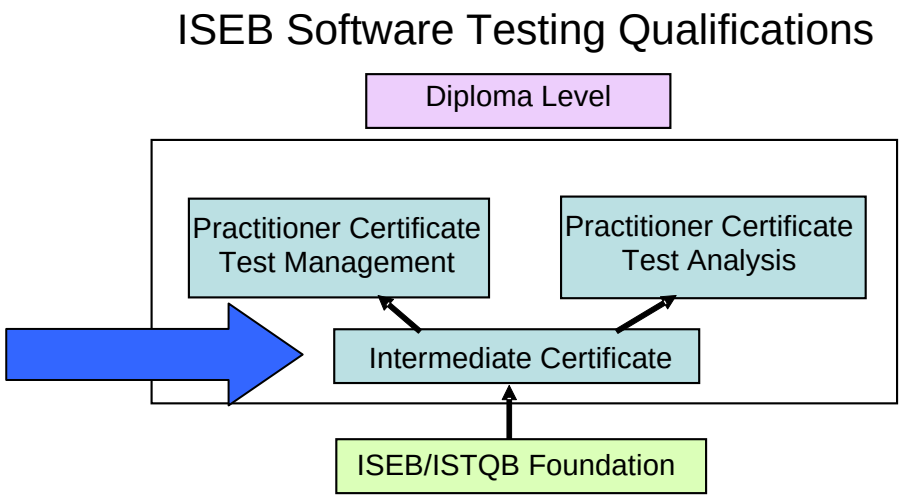
February 2010

Intermediate Certificate in Software Testing

Background

This document is the syllabus for the intermediate paper which leads to the practitioner level of qualification, as administered by BCS Information Systems Examination Board (ISEB). Background information on ISEB and the first level qualification, the Foundation Certificate, may be found at Foundation Certificate in Software Testing.

A testing practitioner is anyone involved in software testing. This includes testers, test analysts, test engineers, test consultants, test managers, user acceptance testers, and software developers.



Within this generic description some practitioners will specialise in the management of the testing function, while others will specialise in the practical testing or test analysis role. Each of these specialisations has its own examination, but both are built on the foundation provided by the intermediate examination defined in this syllabus. The intermediate paper is intended to be an examination appropriate to all testing practitioners. It also constitutes a qualification in its own right. In order to achieve one of the Practitioner Certificates in Software Testing, a candidate must hold the Foundation Certificate and Intermediate Certificate. It is recommended that a candidate sit the Practitioner examination no more than two years after passing the Intermediate level examination. This is to try to ensure that a candidate's knowledge is based upon the published syllabus and terminology.

Entry Criteria

The entry criteria for candidates taking the Intermediate Certificate in Software Testing are as follows:

- hold the ISEB/ISTQB Foundation Certificate in Software Testing

AND EITHER

- have at least 18 months experience in software testing

OR

- have completed an ISEB accredited training course for the Intermediate Certificate in Software Testing

BUT preferably have all three of the above.

The Intermediate Certificate Examination

The examination is a one hour closed book examination (no materials can be taken into the examination room) and will consist of 25 scenario-based multiple choice questions.

The pass mark is 15/25.

Candidates who have a disability or whose first language is not English will normally be allowed an additional 15 minutes extra time. For information on eligibility criteria, please refer to the Reasonable Adjustments Policy on the ISEB website Reasonable Adjustments Policy.

The Intermediate Certificate examination will be based on the syllabus in this document. Examination questions will be drawn from all topics in the syllabus, and coverage of any given topic can be expected to be in proportion to the amount of time allocated to that topic in the syllabus.

Answers to examination questions may require the use of material based on more than one section of this syllabus as well as any material contained in the Foundation syllabus. All sections of the syllabus are examinable.

Objectives of the Intermediate Certificate Examination

The overall objectives of the Intermediate Certificate Examination are:

a) To provide a sound basis for candidates intending to register for either the Test Management Practitioner Examination or the Test Analysis Practitioner Examination. The Intermediate Certificate is a pre-requisite for entry to either of these examinations.

b) To ensure that candidates for either of the Practitioner Certificate Examinations have a broad understanding of the other specialist area.

c) To cover specific topic areas that both specialisations should understand.

The testing skills acquired by attending courses run to this syllabus will be applicable to a wide range of testing challenges that the test practitioner might face. The general approach taught will provide a basis for the testing of software systems in general.

Answers to examination questions may require the use of material based on more than one section of this syllabus or on material from the Foundation syllabus.

Notice to Training Providers

Each major subject heading in this syllabus is assigned an allocated time. The purpose of this is two-fold: first, to give both guidance on the relative proportion of time to be allocated to each section of an accredited course and an approximate minimum time for the teaching of

each section; second, to guide the proportion of questions in the exam. Course providers may spend more time than is indicated and candidates may spend more time again in reading and research. The total time specified is 18 hours of lecture and practical work. Courses do not have to follow the same order as the syllabus. Courses may be run as a single module or broken down into two or three smaller modules.

The syllabus contains references to established standards. The use of referenced standards in the preparation of training material is mandatory. Each standard used must be the version quoted in the current version of this syllabus.

Terminology Used

Terminology used in this document is from the current version of the ISTQB Glossary of Testing Terms or from BS 7925-1, BS 7925-2, IEEE Std. 829-1998, and IEEE Std. 610-1990 as appropriate to the specialist topic. Standards override the glossary in cases of conflict. This syllabus may override both standards and glossary. Versions of standards and glossary change from time to time; the version current at the time of publication of this syllabus is the version referred to in this syllabus as the latest version.

Other Syllabuses

Any references to other ISEB syllabuses refer to the latest published version.

Bloom's Taxonomy

Learning objectives in this syllabus are given indicators from K1-K6. These are based on Bloom's taxonomy of knowledge in the cognitive domain (ref *Taxonomy of Educational Objectives, Handbook 1 – The Cognitive Domain*, Bloom et al., New York 1956), and can be broadly interpreted as follows: K1 – Recall; K2 – Comprehension; K3 – Application; K4 – Analysis; K5 – Synthesis; K6 – Evaluation. Bloom's taxonomy is explained in greater detail in Appendix A, where examples are given. All topics in this syllabus have learning objectives associated with them, each of which has an associated K level. The language used in this syllabus mirrors as closely as possible the language used in defining Bloom's taxonomy to provide candidates with consistent pointers to the expected level of knowledge and a consistent way of expressing that level in words.

Further details and examples of learning objectives are given in Appendix A.

Appendix B contains the rules to which the Practitioner Syllabuses are written.

Appendix C contains references to books and other sources of information to support this syllabus.

Structure of this Document

This document is in three parts. This section deals with the use and purpose of the document itself, and the format of the related examination. The second section of the document contains the syllabus learning objectives with supporting content and commentary. The appendices provide supporting information, covering Bloom's taxonomy of educational objectives, references, and the rules to which this syllabus was written.

This syllabus is structured into sections relating to major subject headings and numbered with a single digit section number. Each section is allocated a minimum contact time for presentation. Learning objectives are identified at the beginning of each section. The K level for each learning objective is identified at the lowest level of breakdown in the learning objectives list.

The breakdown of content matches the structure of the learning objectives, so that the material related to a given learning objective appears in a paragraph bearing the same numerical reference as that of the related learning objective. The content associated with each learning objective may include non-examinable explanatory commentary in italics as well as the examinable content associated with the learning objective.

Change History

Version 1.4	
February 2008	Re-formatted based on new branding guidelines
Version 1.3 June 08	
Section 4.3 vi	The following has been added - Analyse a situation to determine the best estimating approach and make estimates of test effort and duration. (K4) page 18 of the attached document.
Page 26	under the title of Standards, the work “standard” has been corrected in IEEE1044 std.
Version 1.2 – April 2008	
Section 3.1 (ii)	3.3 (ii) on page 15 to be K4
Section 5.1 (ii)	Typo on page 21, 5.1 (ii), line 3. “Show ho**e**” changed to “Show ho**w**”
Version 1.1 – July 2007 First Issue	

Syllabus for Intermediate Certificate in Software Testing

Contents

1. Testing Fundamentals (3 hours)

The topics in this section provide general support for both the Test Management Practitioner and Test Analysis Practitioner Certificate syllabuses and will be assumed in both of these as background knowledge.

Learning Objectives

1.1 Review of the Foundation Certificate Syllabus

i. Recall the main principles and themes from the Foundation syllabus, all of which are considered part of the required knowledge for this syllabus. (K1)

1.2 Application Domains

i. Describe the similarities and differences between typical application domains (K2)
ii. Identify and explain the testing challenges associated with these application domains (K2)
iii. Analyse a situation to determine the testing challenges present in that scenario. (K4)

1.3 Testing in the Life Cycle

i. Recognise and explain the relationship between testing and development (K2)
ii. Identify other processes with which testing interfaces during development (K1)
iii. Explain the relationships between debugging, initial testing during development, confirmation testing and regression testing (K2)
iv. Explain how testing fits into sequential and iterative life cycle models (K2)
v. Describe the testing challenges associated with each life cycle and explain how these challenges can be met (K2)
vi. Analyse a situation to identify the SDLC model(s) in place and select appropriate testing activities to fit with the situation and the life cycle(s) in place (K4)

1.4 Fundamental Test Process

i. Recall the fundamental test process and explain how it may be deployed in different situations and within different life cycle models (K2)

1.1 Review of the Foundation Certificate Syllabus

i. Introduce the issues, philosophy and key points of software testing as covered in material based on the current syllabus of the ISEB/ISTQB Foundation Certificate in Software Testing. Provide an introduction to the ISEB Practitioner Certificates as the next stage in career development.

   In particular, review the following topic areas from the Foundation Syllabus:

   - test levels (component, integration, system, acceptance)
   - test types (targets of testing – functional, non-functional, structural, change-related)

Explain that the four test levels described are not the only possible levels, nor are they necessarily all present in any given life cycle.

1.2 Application Domains

In this syllabus the term 'application domain' refers to groupings of application types that have important similarities in the way hardware and/or software is structured. For example, the client-server domain contains those systems that exhibit a tiered architecture in which one or more components provide services to other (client) components, whereas a mainframe system is characterised as a single large computer providing a service to a large population of users, and PC-based applications provide services to a single user or a small network of users in an office environment. Web-based applications commonly employ a client-server architecture but are separately identified because they have important characteristics arising from their implementation in the context of the World Wide Web. In each case the factors of similarity and difference are those that are expected to impact on the way the application domains are tested, for example the need for special test environments or the applicability of specific test design techniques.

Applications also fall into domains related to the nature of the application itself, as in banking applications or image manipulation applications. Knowledge of the differences between such groups of applications is not required, except in so far as the application has attributes that make it safety critical or business critical. In such cases the implications of the level of criticality of the application on testing should be recognised and identified.

i. Identify the similarities and differences between typical application domains, including web-based applications, client-server applications, large mainframe applications and PC-based applications.

ii. For each application domain type recognise the main testing challenges that must be addressed.

iii. Analyse a situation to identify application domains that may be present and determine the testing challenges associated with the situation.

1.3 Testing in the Lifecycle

The purpose of this section is to elaborate on the coverage of life cycles in the Foundation syllabus to provide a more detailed coverage of the testing challenges associated with different types of life cycle. In this context, testing is relevant to the software development life cycle (SDLC) and also the whole product life cycle, since testing challenges will continue to arise after initial development has been completed. The level of understanding is also raised to K4 by the expectation that a candidate will be able to analyse a situation, identify the nature of any life cycle(s) described within the situation, and recognise the associated testing challenges.

i. Explain how the development process can influence the testing process, for example the object-oriented development paradigm where development makes testing more difficult because of information hiding.

ii. Introduce the concept of interfaces between the test process and other processes, such as project management, configuration management and change management, software development, technical support and technical writing.

iii. Explain the differences between debugging, initial testing, confirmation testing (retesting) and regression testing.

iv. Describe how testing fits into different SDLCs: sequential (e.g. waterfall and V-model), iterative (e.g. Rapid Application Development (RAD), including DSDM and agile methods), and others including the use of Commercial Off the Shelf (COTS) software and constructing software from reusable components.

v. Identify the main differences between the life cycle approaches from a testing perspective and explain the challenges that each life cycle presents to the tester.

vi. Analyse a situation to identify the SDLC model(s) in place and select appropriate testing activities to fit with the situation and the life cycle(s) in place.

1.4 The Fundamental Test Process

i. Recall the fundamental test process as described in the ISTQB Foundation Syllabus and explain how this fundamental test process fits within each of the software development processes (sequential, iterative and COTS-based) described in this section, including how the various activities may be implemented in practice.

2. Reviews (4 hours)

This section has general applicability and will be assumed as background knowledge for both the Test Management and Test Analysis syllabuses, and requires practical examples of the use of reviews. The Test Management syllabus will cover further management issues relating to reviews.

Learning Objectives

2.1 The Principles of Reviews

i. Recall the basic review process as defined in the Foundation syllabus (K1)
ii. Explain the value of using source documents in reviews (K2)
iii. Describe possible outcomes from a review (K2)
iv. Recall different roles that may be defined for a particular review (K1)
v. Recall that reviews may come in a variety of forms and have different objectives (K1)

2.2 Types of Review

i. Describe alternative types of formal review (K2)
ii. Explain the value of informal reviews (K2)

2.3 Using Different Review Types

i. Describe how more than one type of review could be used (K2)
ii. Describe the relationship of reviews to dynamic testing (K2)
iii. Analyse organisations and situations to enable the best choice of review type(s) to be made (K4)

2.4 Performing a Formal Review

i. Provide practical experience of conducting a formal review (K3)
ii. Analyse the effectiveness of the review performed and use the analysis to assess the potential effectiveness of other forms of review in a similar situation (K4)

2.1 The Principles of Reviews

i. Recall the basic review process defined in the Foundation syllabus and recognise the IEEE Std. 1028-1997 Standard for Software Reviews. Explain that, while this syllabus is broadly based on that standard, there is no definitive document that captures current best practice or recognises the many valid variations of review practice that occur in organisations. This syllabus identifies key principles and provides a comparative description of a selection of commonly used review types in the expectation that these will provide an adequate basis for the evaluation, selection and effective implementation of appropriate review techniques.

ii. Recall that reviews are ideally performed when the source documents (those documents that specify the requirements of the product to be reviewed) and those standards to which the product must conform are available. Without source documents, reviews may be limited to finding errors of consistency and completeness within the document under review.

iii. Describe possible outcomes of a review. For example, where a review is conducted as a means of ascertaining the fitness of a work product, there are 3 basic options. First, it may be decided that the product can be used as it is. Second, it may be decided that there are issues that need to be dealt with, but it is unnecessary to perform a further review on the product. Finally, if the error level is found to be too high, then a further review is considered necessary on the product after the issues have been dealt with.

iv. Recall that those involved with reviews often have specific roles and responsibilities. These may include the author of the product, the scribe, who records the issues raised at the review meeting, the chair, who runs the meeting, and the presenter, who leads the meeting through the product. There will often be other reviewers, who simply review the product. There may be a leader for the complete review process. Individual reviewers may be required to look for specific types of issue, and, if so, are often given checklists of issue types to look for.

v. Recall the various forms of review defined in the Foundation syllabus. Recognise that the choice of form of review is dependent on the objectives of the review as well as the culture and maturity of the organisation performing the review.

2.2 Types of Review

This section builds on the coverage of review types in the Foundation syllabus by introducing one new review type and identifying the place of formality in reviews. The material is broadly based on IEEE Std. 1028-1997 but is more contextual in character in that the aim is to develop the capability to make a practical selection of a feasible review type in a given situation rather than to provide a detailed dissertation on reviews.

Evaluation and selection of review techniques relies on knowledge of the organisation and its objectives as well as an understanding of the review types available.

One basic framework that facilitates rational comparison and evaluation of formal review types is contained within IEEE Std. 1028-1997. The description of formal review types in this syllabus uses a simplified version of this framework, which incorporates the following headings:

a) **Introduction** - The purpose and objectives of a review and the typical work products for which it may be used; explain that any type of review requires clear objectives to be defined as an input to the review.
b) **Responsibilities** - The roles and responsibilities associated with the review; explain that these need to be defined according to the type of review and the nature of the source document(s).
c) **Input** - The necessary inputs to the review; explain that these may vary according to the type of review.
d) **Entry Criteria** - The criteria that need to be met before a review can begin; explain that all review types will have some entry criteria, but the nature of the criteria will vary by review type.
e) **Procedures** - Key procedural aspects for a given review; explain how these vary between the types of review.
f) **Exit Criteria** - Criteria to be met before a review can be considered complete; explain the nature of these criteria and how they vary between types of review.
g) **Output** - Typical deliverables for a review type; explain how these vary for each type of review.

i. Compare the different types of formal review and describe their relative strengths and weaknesses and the situations where each may be applicable.

A. Management Review

Explain that:

a) The purposes of management reviews typically include evaluation of management activities and the support of management decisions. Typical products reviewed would include incident reports, risk management plans, test plans.
b) Management reviews will normally include the decision maker for whom the review is being conducted.
c) Inputs will typically include information on the status of the product being reviewed relative to any plan in place for it.
d) Normally management reviews will be scheduled but a review may be held at the request of a decision maker such as the project manager or test manager. Objectives for the review will need to be in place before the review is initiated.
e) Reviewers will be expected to prepare for the review by studying the appropriate documents and providing comments to the decision maker. Rework will be followed up to ensure timely completion.
f) The review is complete when required outputs have been produced and any rework has been defined with an agreed delivery date.
g) Output would normally include a record of any errors and planned actions.

B. Walkthrough

Explain that:

a) Walkthroughs are held to identify errors in documents, but a major objective is commonly to educate reviewers in respect of a project or work product. Source documents would typically include requirements specifications, test strategies, test plans and test specifications.
b) Because walkthroughs are often relatively informal and may address a number of objectives, role assignments may be shared. The author is normally present and walkthroughs are usually restricted to peer groups.

c) The main input is the source document. There may be specifications and standards, but guidelines are often more appropriate.
d) The only entry criterion for a walkthrough would normally be the availability of source documentation and reviewers.
e) Source documents may be distributed before the meeting. Normally the author would make an overview presentation, often using scenarios for requirement and design models and dry runs through code, as a prelude to a general discussion. Reviewers can raise queries or identify errors during the discussion. Errors are normally captured at the walkthrough meeting to enable the author to make corrections after the walkthrough meeting.
f) The walkthrough is considered complete when all reviewers have raised any questions and/or errors and these have been satisfactorily captured.
g) The list of errors may form output from a walkthrough or the source document may be marked up for later correction.

C. Technical Review

Explain that:

a) The purpose of a technical review is normally to evaluate a work product to determine its suitability for intended use and to identify discrepancies from specifications and/or standards. Typical documents reviewed would include requirements specifications and test specifications.
b) Reviewers would normally be staff with appropriate technical expertise to evaluate the source documents. Customer or user representatives or other stakeholders may be invited to attend with a well defined role.
c) As well as the review objectives and source documents, inputs may include relevant standards and specifications.
d) Technical reviews will normally be scheduled, but could also be required by managers, for example to evaluate the impact of incident reports or test results.
e) Thorough preparation is essential for a technical review. Comments should be returned to the review leader and passed on to the author prior to the review meeting. The review leader should confirm that preparation has been adequate before the review begins.
f) The review is complete when the status of the work product is defined and any actions required to meet required standards are defined.
g) Output from a technical review would include a list of any errors and action items related to their correction, and a list of any unresolved issues relating to the work product or any specifications.

D. Inspection

Explain that:

a) The purpose of an inspection is normally to identify and describe errors in a work product in relation to specifications, standards and expected quality attributes. In addition, inspections gather data for the purpose of improving the processes used to create the work product and the inspection process. Typical documents reviewed would include requirements specifications and test specifications.
b) An inspection is led by a trained inspection leader (or moderator). Other reviewers (inspectors) are usually selected to represent different viewpoints and may also be assigned specific review topics (e.g. one may focus on conformance to specification and another on correctness of the design).

c) As well as objectives for the review, inputs normally include appropriate checklists and rule sets for the inspection.
d) Entry criteria are usually very specific and would be expected to include: the work product conforms to appropriate standards for content and format; prior milestones have been achieved; supporting documentation is available.
e) Inspection leaders select inspectors, assign responsibilities and distribute inspection materials before the initial kick-off or overview meeting, which is held so that all participants understand their responsibilities and the product and source documents. At the kick-off, rule sets are identified and inspection rates assigned before inspectors begin their preparation. Each reviewer inspects the product individually and reports back on issues raised in a meeting run by the inspection leader. The meeting focuses on identifying errors, not solving them. The inspection leader collects statistics about the time spent by each participant and the errors found and fixed. An exit decision is made at the meeting.
f) An inspection is considered complete when the output has been defined, any actions have been defined, and required data has been collected.
g) Typical output would be data related to the source document, such as the size of the document, and any errors identified by the inspection, such as the number of errors found by category. Data related to the inspection, such as preparation time and inspection rates would typically also be captured.
h) Explain how one possible outcome of an inspection could be process improvement by sampling a limited number of pages
i) Explain that inspection principles can be applied in an agile context to maximise the benefits of the technique for individuals.

ii. Explain that formality in reviews relates to the following attributes, each of which may be varied: whether or not a meeting is held; nature of entry criteria; level of preparation for meetings; definition of roles, checklists and rule sets of meetings; nature of meetings; extent and nature of output; nature of exit criteria; nature of follow up. Recognise that reviews may be more or less formal. Informal reviews will have the basic characteristics of more formal reviews except in the level of formality. Reviews described in the following subsections relate to formal reviews. Explain that informal reviews can be used as an entry point for organisations with no culture of performing reviews and can facilitate the development of a culture and a discipline that enables more formal review techniques to be implemented when and where appropriate.

2.3 Using Different Review Types

i. Explain how more than one of the review techniques may be employed on a single product. For example, an informal review could be conducted for an early draft of a document, changes could be made as a result of the review, and then a more formal review type could be performed. Compare and contrast the effectiveness of the different review types and identify suitable hybrid types or combinations that would suit particular circumstances. Explain that, in any given situation, a particular type of review, or a sequence of reviews, as described in this syllabus (or as modified by an organisation) will be most appropriate to the specific needs of the situation. Recognise that the ability to make a rational selection, and to identify how the characteristics of the selected review type match the needs of the situation, will determine the effectiveness of the review. Recognise that reviews in real organisations may not be either named or used exactly as described in this syllabus, but that the characteristics and principles described in this syllabus can be used to evaluate the effectiveness of the actual reviews used by an organisation.

ii. Explain why, when reviews are used on code, they should normally be followed by dynamic testing, which can find those types of error that cannot easily be found by static testing.

iii. Analyse development and test organisations and situations and identify an appropriate choice of review type(s) to suit the organisation and the situation.

2.4 Performing a Formal Review

This section is included to ensure that candidates taking an accredited course will have at least one experience of a formal review. Where possible, candidates not taking an accredited (or any other) course should seek this experience before sitting the examination. One key element of this section linked to the practical exercise is the ability to analyse the effectiveness of an actual review activity and use the understanding gained to enhance the activity ot to select a more appropriate review activity for the task in hand.

i. Provide practical experience of a formal review, including planning, individual checking and a review meeting. Analyse the effectiveness of different aspects of the review (e.g. using source documents, using checklists or rulesets, using a slow checking rate).

ii. Analyse the value of the particular type of formal review performed (for example by measuring the time and effort expended for each error found) and draw conclusions about the potential effectiveness of other review types in a similar situation.

3. Testing and Risk (3 hours)

This section provides a general background about risk for the Test Analyst, and will be further developed in the Test Management Syllabus.

Learning Objectives

3.1 Introduction to Risk

i. Recall the nature of product risk and project risk and their effects. (K1)
ii. Explain how risks can interact with other risks. (K2)
iii. Describe typical risks associated with given application domains. (K2)

3.2 Risk Management

i. Recall the core activities of risk management: risk identification, risk analysis and risk mitigation and explain the importance of achieving maximum stakeholder involvement in these activities. (K2)
ii. Explain the relationship between risk and testing. (K2)

3.3 Product Risk Identification and Analysis

i. Identify typical product risks (K2)
ii. Analyse a situation and recognise risks within that scenario. (K4)

3.1 Introduction to Risk and Risk-Based Testing

i. Review the definition of risk and the difference between project and product risks. Explain that project risks primarily affect the successful completion of the project (e.g. extend the delivery timescales) and are addressed by project management actions. Explain that product risks primarily impact the product (e.g. a software failure that takes the delivered system out of operation) and are important drivers of testing strategy.

ii. Explain that project risks and product risks will affect each other: a project risk, by causing delay, may impact on the quality of work and so lead to additional product risk; a product risk (e.g. a complex design) may cause extra testing to be incorporated, leading to delay and associated project risk.

iii. Describe typical risks (both product and project) to be considered such as those related to safety, economic, security, political and technical factors and give examples of each of these. Relate risks to application domains and identify typical risks to be expected in a given domain.

Project risks and risk mitigation will be covered in more detail in the Test Management syllabus.

3.2 Risk Management

The term' stakeholder' is used in this syllabus to identify any person with an interest in the outcome of a project. This will normally include software developers and testers as well as users and managers. Where appropriate stakeholders will be identified by the stakeholder group (e.g. testers) to which they belong rather than as individuals.

i. Explain that ideally all stakeholder groups should be directly involved at all stages of risk management.

ii. Give examples of how project and product risks can be addressed by decisions documented in the test strategy, such as which test levels to include, and which test case design techniques and test exit criteria to use.

3.3 Product Risk Identification and Analysis

i. Identify product risks that could be addressed by testing (e.g. a risk of slow response times could be addressed by performance testing).

ii. Analyse a situation to determine possible risks associated with the application domain, the life cycle and the development approach.

4. Test Management (4 hours)

This section provides a basic introduction to test management that will be developed in greater detail in the Test Management syllabus and will provide essential background for the Test Analysis syllabus.

Learning Objectives

4.1 Test Policy, Test Strategy, Test Plans

i. Explain the hierarchy of test management and their associated documentation (K2)
ii. Describe the role and purpose of each document in the test management hierarchy (K2)
iii. Analyse a suite of documentation to determine its effectiveness in defining policy, strategy and plans (K4)

4.2 Test Entry and Exit Criteria

i. Explain the significance of objective test entry and exit criteria (K2)
ii. Give examples of suitable test entry and exit criteria and explain possible alternative courses of action when test entry and exit criteria are not met (K2)
iii. Analyse testing situations to select appropriate test entry and exit criteria (K4)

4.3 Estimating Techniques

i. Explain the nature and importance of estimation applied to testing (K2)
ii. Explain the methods and inputs/parameters available to estimate the time and effort required to design, document, schedule and execute a test or a collection of tests (K2)
iii. List different estimation methods, and the value of using more than one method (K2)
iv. Explain the difference between an estimate and a target (K2)
v. Explain why more than one cycle or iteration of test execution should be estimated (K2)
vi. Analyse a situation to determine the best estimating approach and make estimates of test effort and duration (K4)

4.4 Test Monitoring

i. Describe how testing may be monitored, and give examples of what could be monitored (K1)
ii. Identify and describe typical measures of test progress and test quality (K2)
iii. Identify the content of test summary reports appropriate to a range of stakeholders and at different stages of the test process and at different points in the life cycle (K2)
iv. Analyse a test summary report to assess the testing reported on and decide on required control actions (K4)

4.5 Incident Management Process

i. Describe how incidents are reported, tracked and analysed to ensure that remedial action is effective (K2)
ii. Describe alternative incident management processes (K2)
iii. Analyse a simple incident management process and determine what improvements could be made (K4)

4.1 Test Policy, Test Strategy, Test Plans

i. Explain the hierarchy of test management documentation, including test policy, test strategy, high level or project test plan and level test plan.

ii. Identify the relationships between the various levels of document and explain the purpose of each level. Explain that there are alternative ways to document test management information and that documents will not necessarily have the names used in this syllabus.

iii. Analyse document hierarchies and documents to determine whether policy, strategy and planning issues are adequately documented.

4.2 Entry and Exit Criteria

i. Explain the concept and purpose of overall test project entry and exit criteria (sometimes known as acceptance criteria or test completion criteria), and of entry and exit criteria for test levels. Explain the difference between test entry and exit criteria, including the possibility that, as well as exit criteria from the previous test level, entry criteria may depend on factors such as the availability of a test environment for the phase being entered.

ii. Give examples of typical test entry and exit criteria, and explain what alternatives are available when they are not met, and explain that the alternatives depend on the organisation's approach to risk management and on overall organisational (test) policy.

iii. Analyse testing situations and select appropriate test entry and exit criteria to meet defined objectives.

4.3 Estimating Techniques

i. Explain that an estimate is an approximate calculation or judgement, typically based on the professional understanding of experienced practitioners. Estimates are needed for the time and effort required to carry out test activities as described in a test plan. They are important in determining the correct type, level and number of resources required and the expected duration of the planned testing activities.

ii. Estimates should take account of dependent activities, such as the elapsed time for fixing defects found in test execution, and the time and effort needed for subsequent confirmation testing and regression testing.

iii. Explain the following methods of reaching an estimate:
- Intuition or guessing
- Based on metrics from previous testing efforts
- Formula based, using a technique such as Test Point Analysis (TPA) or a standard percentage of development time
- Consensus of knowledgeable people
- Detailed micro-estimates of all activities in a work breakdown structure

Estimates can be carried out:

- Top down, e.g. based on the percentage of development effort or previous total test efforts
- Bottom up, e.g. based on estimating each task in a work breakdown structure

Explain the importance of deriving estimates by more than one method, if possible, to increase confidence in the estimate. Explain also the value of using more than one estimator and working towards a consensus on the estimate, and the importance of capturing data to determine how accurate the previous estimates were, to allow feedback to be given to improve future estimates.

iv. Explain that deadlines represent targets that are desirable for political or commercial reasons, but estimates are based on a given scope of work.

v. Explain that several cycles (or iterations) of running tests will normally be required (run tests, report defects, confirmation tests, running of any tests that were not able to be run in the first cycle, and regression testing).

vi. Analyse testing situations to determine the best estimating approach and make estimates of required test effort and duration, including estimates of expected levels of confirmation and regression testing.

4.4 Test Monitoring

i. Explain that testing is monitored in order to compare progress with plans. Typical elements that may be monitored include:

- test preparation work completed (compared to planned work)
- tests run versus the planned test set
- result of completed tests (pass/fail) versus planned tests
- number of incident reports raised, and ranked by severity and priority
- time and/or effort taken to perform test activities (e.g. setting up a test environment)
- coverage of the system or software by tests run

ii. Identify and describe typical measurements used to quantify test progress and test quality, including measures of defect density and the capture of defect trends, for example by drawing graphs of test activity shown against a time line. Factors affecting the testing activities should be reported, such as a test environment not being available, confirmation tests failing, or more defects found than expected. Describe how information and metrics from test monitoring may be used for test process improvement.

iii. Describe the typical contents of a test summary report from a tester to a test leader/test manager for different test levels and different SDLCs.

iv. Analyse a test summary report to identify the quality and completeness of testing reported on.

4.5 Incident Management Process

i. Describe the incident life cycle: Recognition, Investigation, Action, and Disposition, as described in IEEE 1044. Illustrate the differences between a simple and a more complex incident process. Explain the purpose of incident reporting as a mechanism for identifying potential issues with the software under test and describe the possible outcomes of incident analysis. Explain that incidents can be raised for software, tests or documentation.

ii. Describe mechanisms for tracking and analysing incidents and explain the importance of ensuring that all incidents are tracked until some resolution is achieved. Identify alternative resolutions for an incident. Give examples of tracking mechanisms (e.g. a spreadsheet or a defect tracking tool).

iii. Analyse a simple incident management process to determine its effectiveness and identify any improvements that could be made.

5. Test Analysis (4 hours)

This section provides a basic introduction to test analysis that will be developed in greater detail in the Test Analysis syllabus and will provide essential background for the Test Management syllabus.

Learning Objectives

5.1 Fundamentals of Test Analysis

i. Define the test analysis function (K2)
ii. Recall the basic relationships between test basis, test condition, test case and test procedure (manual test script) (K1)

5.2 Test Environment Requirements

i. Identify and explain the principles behind determination of test environment needs for executing tests (K2)
ii. Analyse a situation to determine test environment requirements (K4)

5.3 Selection of Techniques

i. Recognise alternative approaches to testing, including the fundamental differences between static and dynamic testing and between scripted and unscripted testing, and describe the strengths, weaknesses and appropriate uses of each (K2)
ii. Describe the categories of test techniques available to testers (K2)
iii. Describe possible criteria for selecting test design techniques (K2)
iv. Explain the benefits and pitfalls of deploying test design techniques (K2)
v. Analyse a practical testing situation and select appropriate test design techniques (K4)

5.4 Coverage Measures

i. Explain the concept of coverage and recognise and define suitable measures of coverage (K2)
ii. Explain the importance of defining what coverage measures mean in a practical situation (K2)
iii. Analyse a practical testing situation and select appropriate coverage measures (K4)

5.1 Fundamentals of Test Analysis

i. Explain the typical activities of test analysis as including: analysis of test basis documents; generation of test conditions; determination of suitable test cases; creation of test procedures, and explain that test analysis includes the selection of the most suitable test design techniques and their effective deployment and the use of appropriate measures to quantify completeness of testing.

ii. Explain how test bases can be constructed from existing documentation or created from scratch when no documentation exists, and describe how test conditions can be extracted and analysed to generate test cases using suitable test design techniques. Show how test procedures (manual test scripts) or automated test scripts can be constructed from test conditions and test cases.

5.2 Test Environment Requirements

i. Explain the nature and purpose of a test environment and give examples where complex test environments may be needed and will need to be planned early in the life cycle (e.g. testing of an interactive system intended for continuous operation in an organisation with offices in several continents).

ii. Analyse an outline development or maintenance project plan and identify the generic test environment requirements associated with testing in that situation.

5.3 Selection of Techniques

i. Explain the nature and purpose of static testing and dynamic testing. Describe where each can be used in the life cycle and how they can complement each other. Give examples of each. Explain the advantages and disadvantages of scripted testing over unscripted testing (e.g. ad hoc or exploratory testing) and give examples of situations where each would be the preferred approach. Describe how the approaches are complementary to each other and how they can be effectively combined.

Dynamic testing, scripted testing and unscripted testing will be developed in greater detail in the Test Analysis syllabus.

ii. Explain the role and purpose of test design techniques and recall the categories of techniques that are available: specification-based, structure-based and experience-based.

iii. Identify and explain typical criteria that should be applied when selecting test design techniques for a test effort.

iv. Describe the main benefits of using test design techniques and the main pitfalls in implementing test design techniques in a project.

v. Analyse a test basis for an application and select suitable combinations of static and dynamic testing, scripted and unscripted testing, and test design techniques.

5.4 Coverage Measures

This topic will be developed in greater detail in the Test Analysis syllabus.

i. Explain the principle of coverage measurement and show how it can be used to define an objective exit criterion for testing. Give examples of alternative coverage measures at each test level (e.g. statements or decision outcomes at component level, requirements coverage at user acceptance testing level). Explain that coverage measures can be applied to functional, non-functional and structural testing. Describe how tests are derived by identifying functional non-functional or structural items not yet tested (covered) by a test suite, and explain that test design is normally done to increase coverage of the software or system. Explain the role of static analysis in providing structural data from which to derive structural tests. Explain the need for appropriate mechanisms, such as instrumentation, to enable achieved coverage to be measured at test execution. Identify measures with respect to test progress and completeness, such as test completion percentage (tests run / tests planned to be run) and test success percentage (tests that pass / tests run). See also Section 4.4 Test Monitoring. Explain that the term 'test coverage' is sometimes used inappropriately to refer to percentages of tests completed as well as to code or system coverage percentages.

ii. Explain that coverage measures must define which elements or non-functional attributes of a component or system have been covered by tests.

iii. Analyse a test basis for an application and select appropriate coverage measures for use at different test levels within the project.

Appendix A: Learning Objectives/Levels of Knowledge

The following learning objectives are defined as applying to this syllabus. Each topic in the syllabus will be examined according to the learning objective for it.

Level 1: Remember (K1)

The candidate will recognise, remember and recall a term or concept.

Example

Can recognise the definition of “failure” as:

- “non-delivery of service to an end user or any other stakeholder” or
- “actual deviation of the component or system from its expected delivery, service or result”.

Level 2: Understand (K2)

The candidate can select the reasons or explanations for statements related to the topic, and can summarise, compare, classify and give examples for the testing concept.

Examples

Can explain the reason why tests should be designed as early as possible:

- To find defects when they are cheaper to remove.
- To find the most important defects first.

Can explain the similarities and differences between integration and system testing:

- Similarities: testing more than one component, and can test non-functional aspects.
- Differences: integration testing concentrates on interfaces and interactions, and system testing concentrates on whole-system aspects, such as end to end processing.

Level 3: Apply (K3)

The candidate can select the correct application of a concept or technique and apply it to a given context.

Examples

- Can identify boundary values for valid and invalid partitions.
- Can select test cases from a given state transition diagram in order to cover all transitions.

Level 4: Analyse (K4)

The candidate can separate information related to a concept or technique into its constituent parts for better understanding, and can distinguish between facts and inferences.

Examples

- Can understand the various options available for risk identification.
- Can describe which portions of an incident report are factual and which are inferred from the results.

Level 5: Synthesise (K5)

The candidate can identify and build patterns in facts and information related to a concept or technique, and can create new meaning or structure from parts of a concept.

Examples

- Can design a quality risk analysis process that includes both rigorous and informal elements.
- Can create a blended test strategy that uses a dynamic strategy to balance an analytical strategy.
- Can combine aspects of different review processes to form an effective process for their organisation.

Level 6: Evaluate (K6)

The candidate can judge the value of information and decide on its applicability in a given situation.

Examples

- Can determine the relative effectiveness and efficiency of different review processes or different testing techniques.
- Can determine the type of information that should be gathered for an incident report.

References

(For the cognitive levels of learning objectives)

Bloom, B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain, David McKay, Co. Inc.

Anderson, L. W. and Krathwohl, D. R. (eds) (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon.

Appendix B: Rules for the ISEB SWT Intermediate Level Syllabus

Syllabus General Rules

SG1 – The syllabus should be understandable and absorbable by people who hold the ISEB/ISTQB Foundation Certificate.

SG2 – The syllabus should be more practical than theoretical.

SG3 – The syllabus should be clear and unambiguous to its intended readers.

Syllabus Content

SC1 – The syllabus should include recent testing concepts and should reflect current best practice in software testing where this is generally agreed and has been published. The syllabus is subject to review every three to five years.

SC2 – The syllabus should minimise time-related issues, such as current market conditions, to enable it to have a shelf life of three to five years.

SC3 – Each statement in the Syllabus should clearly state what the candidate is expected to know and therefore what can be examined. (E.g. “Explain that the criteria are” not “Explain the criteria for”).

Learning Objectives

LO1 – Learning objectives should distinguish between applicable knowledge levels according to Bloom’s taxonomy (K1 to K6).

LO2 – The description of the content should be consistent with the learning objectives.

LO3 – To illustrate the learning objectives, sample exam questions for each major section should be issued along with the syllabus.

LO4 – Each section of the syllabus should include a K-level.

Structure rules

ST1 – The structure of the syllabus should be clear and allow cross-referencing to and from other parts, from exam questions and from other relevant documents.

ST2 – Overlap between sections of the syllabus should be minimised. Overlap between related syllabuses (Foundation, Intermediate, Analyst & Management) should be minimised, or stated (if intentional).

ST3 – Each syllabus and each section of each syllabus should have the same structure and format.

ST4 – The syllabus should contain version, date of issue and page number on every page.

ST5 – The syllabus should include a guideline for the amount of time to be spent in each major section (to reflect the relative importance of each topic).

ST6 – Each statement should be consistent with the ISTQB Foundation Syllabus (where it covers the same area) and should use the same terminology.

Syllabus References

SR1 – Sources and references should be given for concepts in the syllabus to help training providers find out more information about the topic.
SR2 – Where there are not readily identified and clear sources, more detail should be provided in the syllabus. For example, definitions are in the Glossary, so only the terms are listed in the syllabus.

Appendix C: Syllabus References

References and other texts listed on this page are relevant to the Software Testing Practitioner syllabus as a whole. There may be further references for the Practitioner in Test Management and Practitioner in Test Analysis syllabuses.

Syllabuses

[ISTQB F Syllabus] ISTQB Certified Tester: Foundation Level Syllabus (Version 1/7/2005)
[ISTQB Glossary] ISTQB - Veenendaal, Erik van (ed.) (2006), Standard glossary of terms used in Software Testing, Version 1.2
ISEB Practitioner Certificate in Software Test Analysis (under development)
ISEB Practitioner Certificate in Software Test Management (under development)

Web Sites

ISEB Web site: www.iseb.org.uk
ISTQB Web site: www.istqb.org

Standards

IEEE 1028 (1998), Standard for Software Reviews, IEEE Standards Board
ISO/IEC 12207 (1995), Information Technology – Software Life Cycle Processes
[IEEE 829] IEEE Std 829™ (1998/2005) IEEE Standard for Software Test Documentation (currently under revision)
[ISO 9126] ISO/IEC 9126-1:2001, Software Engineering – Software Product Quality
[IEEE 1044] IEEE Std. 1044-1993, Standard Classification for Software Anomalies, IEEE Std. 1044.1-1995, Guide to Classification for Software Anomalies.
[BS 7925-1] BS 7925-1:1998, Software Testing Part 1: Vocabulary
[BS 7925-2] BS 7925-2:1998, Software Testing Part 2: Software Component Testing
[IEEE 610] IEEE Std 610 (1990), IEEE Standard Computer Dictionaries

Books

Note that these may not be directly referenced in this Syllabus, but represent a selection of published works that support the principles given in this Syllabus.

[Bach] James Bach (2004), Exploratory Testing, in: Erik van Veenendaal, The Testing Practitioner –2nd edition, UTN Publishing, ISBN 90-72194-65-9.

[Beizer] Beizer, Boris (1990) Software Testing Techniques (2nd edition), Van Nostrand Reinhold: Boston.

[Black, 2001] Black, R. (2001) Managing the Testing Process (2nd edition), John Wiley & Sons: New York.

[Black, 2004] Black, R. Critical Testing Processes, Addison Wesley: Reading MA.

[Copeland] Copeland, Lee (2004) A Practitioner's Guide to Software Test Design, Artech House: Norwood, MA.

[Craig] Craig, Rick D. and Jaskiel, Stefan P. (2002) Systematic Software Testing, Artech House: Norwood, MA.

[Drabick]Drabick, R., Best Practices for the Formal Software Testing Process: A menu of Testing Tasks, Dorset House.

[Freedman and Weinberg] Daniel Freedman and Gerald Weinberg (1990), Walkthroughs, Inspections, and Technical Reviews, Dorset House Publishing, ISBN 0-932633-19-6.

[Gerrard] Paul Gerrard and Neil Thompson (2002), Risk-Based E-Business Testing, Artech House Publishers, ISBN 1-58053-314-0.

[Gilb and Graham] Tom Gilb and Dorothy Graham (1993), Software Inspection, Addison-Wesley, ISBN 0-201-63181-4.

[Hambling] Brian Hambling (Ed.), Peter Morgan, Angelina Samaroo, Geoff Thompson, Peter Williams (2006), Software Testing: An ISEB Foundation, British Computer Society, Swindon.

[Hetzel] William (Bill) Hetzel (1988), The complete guide to software testing – 2nd edition, QED Information Sciences, ISBN 0-89435-242-3.

[Kaner] Kaner, Cem, Bach, James and Pettticord, Bret (2002) Lessons Learned in Software Testing, John Wiley & Sons.

[Perry] Perry, W E and Rice, R W., (1997) Surviving the Top-Ten Challenges of Software Testing: A People Oriented Approach Dorset House.

[Myers] Glenford Myers (1979), The Art of Software Testing, 2nd Ed., Wiley, ISBN 0-471-46912-2.

[TMap] Martin Pol, Ruud Teunissen, Erik van Veenendaal (2002), Software Testing, A guide to the TMap Approach, Addison Wesley, ISBN 0-201-745712.

[Veenendaal] Erik van Veenendaal (2004), The Testing Practitioner – 2nd edition, UTN Publishing, ISBN 90-72194-65-9.