BCS Foundation Certificate in Systems Development Syllabus

Version 2.7
December 2017

This qualification is not regulated by the following United Kingdom Regulators - Ofqual, Qualification in Wales, CCEA or SQA
BCS Foundation Certificate in Systems Development Syllabus

Introduction ........................................................................................................................ 4

Objectives .......................................................................................................................... 4

Course Format and Duration .............................................................................................. 5

Eligibility for the Examination ............................................................................................. 5

Format of the Examination ................................................................................................. 5

Additional time ................................................................................................................... 5

Excerpts from BCS books .................................................................................................. 6

Guidelines for Accredited Training Organisations ............................................................... 6

Use of Calculators .............................................................................................................. 6

Syllabus ............................................................................................................................. 7

1. Introduction to Systems Development (5%, K2) 7
2. Lifecycle types and their rationales (10%, K2) 7
3. Business Analysis (5%, K2) 8
4. Requirements Engineering (10%, K2) 8
5. Making a Business Case (10%, K3) 9
6. Programming and Development Approaches (10%, K2) 10
7. Systems Modelling and Specification Techniques (10%, K2) 10
8. Systems Design (10%, K2) 11
9. System Architecture (5%, K2) 12
10. Quality and Testing (10%, K2) 13
11. Implementation and Changeover (5%, K2) 14
12. Evaluation and Maintenance (5%, K2) 15
13. Software Support Tools (5%, K2) 16

Recommended Reading .................................................................................................. 18

Levels of Knowledge / SFIA Levels .................................................................................. 19

Question Weighting .......................................................................................................... 19

Format of Examination ..................................................................................................... 20

Trainer Criteria ................................................................................................................ 20

Classroom Size ................................................................................................................ 20
### Change History

Any changes made to the syllabus shall be clearly documented with a change history log. This shall include the latest version number, date of the amendment and changes made. The purpose is to identify quickly what changes have been made.

<table>
<thead>
<tr>
<th>Version Number and Date</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 2.7 Dec 2019</td>
<td>Inclusion of Recommended Reading</td>
</tr>
<tr>
<td>Version 2.6 March 2017</td>
<td>Standardised new template format adopted, with revised ToC. K levels checked and Reasonable Adjustments text changed. No material change to syllabus content.</td>
</tr>
<tr>
<td>Version 2.5 December 2016</td>
<td>Strapline regarding regulated statement has been added</td>
</tr>
<tr>
<td>Version 2.4 March 2015</td>
<td>Updated language requirements for additional time and use of dictionaries. Added trainer criteria and classroom size.</td>
</tr>
<tr>
<td>Version 2.3 November 2012</td>
<td>Updated Reasonable Adjustments for foreign language candidates</td>
</tr>
<tr>
<td>Version 2.2 September 2012</td>
<td>Updated additional time for foreign language candidates and included a section to cover excerpts from BCS books</td>
</tr>
<tr>
<td>Version 2.1 August 2012</td>
<td>Added in details of extra time for foreign language candidates. Added in Levels of Knowledge and Skills Levels and Learning Outcomes.</td>
</tr>
<tr>
<td>Version 2.0 May 2012</td>
<td>Logos and footers updated. Reference to ISEB removed throughout document. Document control sheet updated. Learning hours added to the Format of the Examination Sections. No change to the content of the syllabus.</td>
</tr>
<tr>
<td>Version 1.1</td>
<td>Re-branded and re-formatted. Added in a contents page, change history and the examination format on final page.</td>
</tr>
</tbody>
</table>
Introduction

This BCS Foundation Certificate in Systems Development is designed for anyone involved in or affected by the development of IS/IT systems: this extends to Business and Systems Analysts, Designers, Developers, Testers and other users and practitioners who want an understanding of the coverage of Systems Development. Candidates will be able to demonstrate an understanding of the principles systems development and delivery, including, life cycle approaches, architecture, business analysis, requirements engineering, systems modelling, design, development, testing, implementation and communication between the various roles involved in systems development and delivery.

This exam sits below the range of BCS modular Certificates in Systems Development and the BCS Diploma in Solution Development. It can be used to provide a foundation in the subject for specialists in other disciplines.

A pass in this Certificate is an optional requirement of the BCS International Diploma in Systems Development, for candidates wishing to follow that certification pathway later.

Objectives

This course provides candidates with an understanding of the principles of, and practical experience of using, industry best practice involved in Systems Development.

Specific Learning Objectives

A comprehensive range of topics are covered, including:

- Introduction to Systems Development
- Life Cycle types and their Rationales
- Business Analysis
- Requirements Engineering
- Making a Business Case
- Programming and Development Approaches
- Systems Modelling and Specification Techniques
- Systems Design
- System Architecture
- Quality and Testing
- Implementation and changeover
- Evaluation and maintenance
- Software Support Tools
Course Format and Duration

Candidates can study for this certificate in two ways: by attending training courses provided by an Accredited Training Organisation or by self-study. An accredited training course will require a minimum of 21 hours of study run over a minimum of three days.

The course can be delivered a number of different ways from traditional class-room based training to online e-learning.

Eligibility for the Examination

There are no specific pre-requisites for entry to the examination; however candidates should possess the appropriate level of knowledge to fulfil the objective shown above.

Format of the Examination

- 60 minute ‘closed book’
- 40 multiple choice questions
- Pass mark is 26/40 (65%)

The 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.

Additional time

For candidates requiring reasonable adjustments

Please refer to the reasonable adjustments policy for detailed information on how and when to apply.

For candidates whose language is not the language of the examination

If the examination is taken in a language that is not the candidate’s native/official language, candidates are entitled to:

- 25% extra time
- Use their own paper language dictionary (whose purpose is translation between the examination language and another national language) during the examination
  Electronic versions of dictionaries will not be allowed into the examination room
Excerpts from BCS books

Accredited Training Organisations may include excerpts from BCS books in the course materials. If you wish to use excerpts from the books you will need a license from BCS to do this. If you are interested in taking out a licence to use BCS published material, you should contact the Head of Publishing at BCS outlining the material you wish to copy and the use to which it will be put.

Guidelines for Accredited Training Organisations

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; secondly, to guide the proportion of questions in the exam. Accredited Training Organisations may spend more time than is indicated and candidates may spend more time again in reading and research. 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.

Note that specific laws and legal issues relating to the country(s) within which a training provider operates may be mentioned as examples and included in course material, but the examination will only test the principles.

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.

Use of Calculators

No calculators or mobile technology will be allowed.
Syllabus

For each top-level area of the syllabus a percentage and K level is identified. The percentage is the exam coverage of that area, and the K level identifies the maximum level of knowledge that may be examined for that area.

1. Introduction to Systems Development (5%, K2)

The objective is to understand the scope of systems development work and its relationship to other associated disciplines.

1.1 What is systems development
1.2 The scope of systems development
1.3 Relationship with other disciplines such as project management, programming, testing, service management, change and configuration management

2. Lifecycle types and their rationales (10%, K2)

The objective is to understand the range of systems development lifecycles, their application, advantages and disadvantages.

2.1 Systems Development Lifecycles
   - Waterfall model
   - V model
   - Incremental model
   - Spiral model
   - Unified Process

2.2 For each lifecycle
   - Principles and rationale
   - Structure and stages
   - Advantages
   - Disadvantages
   - Selection criteria
   - Team roles and responsibilities

2.3 Adaptation and customisation of the lifecycles

2.4 Project management and the lifecycles
   - Responsibilities of project managers in systems development
   - Difference between project life cycles and systems development life cycles
3. Business Analysis (5%, K2)

The objective is to understand the objectives, activities and deliverables of business analysis work.

3.1 Definition of Business Analysis

3.2 Holistic discipline
- Focus on business problems and opportunities
- Bridge between business and IT

3.3 Place of Business Analysis in the development lifecycle
- Feasibility study
- Requirements analysis
- User acceptance testing
- Implementation
- Post-implementation review/benefits realisation

3.4 Outcomes from Business Analysis
- People change
- Process change
- IT/IS change
- Organisation change

4. Requirements Engineering (10%, K2)

The objective is to understand the core activities of the requirements engineering approach and the techniques used to define, document and manage requirements.

4.1 Key areas of Requirements Engineering
- Requirements elicitation
- Requirements analysis
- Requirements negotiation
- Requirements documentation
- Requirements validation

4.2 Techniques for requirements elicitation
- Workshops
- Interviews
- Observation
- Questionnaires
- Scenarios
- Prototyping
- Document analysis

4.3 For each elicitation technique
- Purpose of the technique
- Advantages
4.4 Types of requirements
   - Functional
   - Non-functional

4.5 Prioritising requirements
   - Reasons for prioritisation
   - Approach to prioritisation

4.6 Managing requirements
   - Recording requirements documentation
   - Change control
   - Version control
   - Traceability
   - CASE tools

4.7 Analysing and validating requirements
   - Feasibility checking
   - Ensuring the consistency and correctness of the requirements
   - The validation process and roles
   - Responsibilities of the reviewers

5. Making a Business Case (10%, K3)

The objective is to understand the purpose of producing a business case and the structure and contents of a business case.

5.1 Feasibility checking
   - Business feasibility
   - Technical feasibility
   - Financial feasibility

5.2 Elements of a business case
   - Background and context to the business case
   - Options
   - Costs and benefits for each option
   - Impacts of each option
   - Risks of each option
   - Recommended actions

5.3 Identifying, evaluating and selecting options

5.4 Principles of cost/benefit analysis

5.5 Principles of impact and risk analysis
6. **Programming and Development Approaches (10%, K2)**

The objective is to appreciate the different approaches to programming and development of software solutions and identify the key features of each.

6.1 Types of development approaches (these are not necessarily mutually-exclusive)

- Agile Approaches
  - Extreme programming
  - Iterative and incremental approaches (Dynamic Systems Development Method)
  - Exploratory/empirical approaches (Scrum, Adaptive Systems Development, Crystal, Lean Development)
  - Feature driven development
  - Test driven development
- Procedural
  - Waterfall lifecycle
  - Structured Programming
- Object-Oriented approaches
- Service Oriented
  - On Demand Software
  - Application Service Provider (ASP)
- Open source development
- Commercial off-the-shelf packages (COTS)

7. **Systems Modelling and Specification Techniques (10%, K2)**

The objective is to understand the importance of modelling and documentation in the systems development process, to identify the different types models and be aware of the various perspectives they address.

7.1 Reasons for modelling

- To aid communication between actors
- As a basis for rigorous development
- To provide a standard approach
- To ensure consistency across the development
- To assist in the identification of re-use
- To compare the current situation with the required

7.2 Modelling from different perspectives

- Examples of models
- Modelling perspectives of ‘Why; What; How; When; Who; Where’
- Modelling static data (top down and bottom up)
- Modelling process and business rules
- Modelling dynamic behaviour
- Modelling user interface
7.3 Cross-referencing different modelling perspectives
- Cross-referencing process and data (e.g. CRUD)
- Cross-referencing process and objectives
- Cross-referencing user roles and processes

7.4 Documentation and specification
- Importance of documentation
- Documentation configuration management and version control
- Keeping documentation up to date

8. Systems Design (10%, K2)
The objective is to recognise the fundamental objectives and principles of good systems design.

8.1 The location of systems design in the systems development lifecycle

8.2 The objectives and constraints of systems design
- Objectives reflect many of the software qualities also identified in section 10. They include the need to deliver required functionality, reliability, maintainability, flexibility, expandability, usability, efficiency, re-usability, testability and adherence to standards
- Constraints on design include budget, time, skills available, influence of current (legacy) systems, target hardware and software platforms and internal politics

8.3 Input design, input technologies and their application
- Keyboard input considerations including data validation and data verification requirements
  - Existence check
  - Range check
  - Format check
  - Cross-field (consistency) check
- Cost, time and accuracy advantages of direct data input
- Direct data input technologies including signals, voice, scanning, Optical Character Recognition (OCR), Magnetic Ink Character Recognition (MICR), Optical Mark Recognition (OMR), bar codes, swipe cards and mouse
- Application of direct data input technologies within given scenarios

8.4 Output technologies and their application (screen and form design is covered in the interface design and usability section of the syllabus)
- Output design technologies including different types of screens, printers and digital media, public display screens (for example, at bus stops) and mobile telephones
- Application of output design technologies within given scenarios
8.5 The objectives and principles of process design
- Stepwise refinement – the process of the elaboration of requirements
- Expressing processes through the constructs of sequence, condition and repetition
- Modularity and the principles of coupling and cohesion
- The principles of abstraction, encapsulation and generalisation

8.6 The objectives and principles of data design
- The aims and principles (but not conduct) of normalisation
- File organisation methods: serial, sequential, index-sequential and random
- File access methods: searches, indexes and algorithms
- Principles of hierarchical, network, relational and object-oriented database management systems

8.7 The design of codes
- Factors affecting the design of a successful code, including uniqueness, stability, expandability and length
- The principles of facetted codes
- The use and definition of check digits

8.8 The scope and principles of security design
- Physical security of the computer environment
- Logical security measures such as passwords
- Firewalls, anti-virus software and spy-ware
- The function and content of audit trails
- Principles of the Computer Misuse Act
- Principles of the Data Protection Act

9. System Architecture (5%, K2)

The objective is to recognise the importance of a well-developed architecture as an integral part of good system development.

9.1 Types of architecture
- Enterprise architecture and IT architecture
- Systems and application architecture
- Data architecture

9.2 Objectives and principles of systems architectures
- What is an IT architecture?
- Why is an architecture important?
- Underlying principles of architectures

9.3 Stakeholders and roles in architecture
- Customers and sponsors
- Service and product providers
Designers and developers
IT architect

9.4 Management of the architecture
- Monitoring compliance
- Handling change
- Evolution of architecture

9.5 The tiered architecture approach to IT system development
- The components of an IT architecture
- Interface, process and data layers

9.6 Service Oriented Architecture and Service Oriented Development Applications
- Services and technologies
- Examples of services

10. Quality and Testing (10%, K2)

The objective is to recognise the role of testing through the lifecycle.

10.1 The definition of software quality
- The need to meet robustness and reliability requirements
- The need to meet functional requirements
- The need to meet non-functional requirements, particularly usability
- The need for inherent software product qualities such as maintainability, flexibility and efficiency

10.2 The objectives and limitations of testing
- The causes of software defects and the distinction between errors, faults and failures
- The distinction between static and dynamic testing
- General testing principles
  - Testing shows the presence of defects
  - Exhaustive testing is impossible
  - The principle and benefits of early testing
  - The recognition of defect clustering

10.3 The structure and purpose of the static test stages of the V model
- Requirements
- Functional Specification
- Design Specification
- Module specification

10.4 The purpose and content of the dynamic test stages of the V model
- Component (unit) testing
- Component integration testing
- System testing
- Functional testing
- Non-functional testing
- User acceptance testing

10.5 Static testing
- Reviews and the test process
- Informal review
- Walkthrough
- Technical review
- Inspection

10.6 Dynamic testing
- Specification-based or black-box techniques
- Structure-based or white-box techniques
- Experience-based techniques

10.7 Re-testing (confirmation testing)

10.8 Regression testing
- Definition and scope of regression testing
- Opportunities for automating regression testing

11. Implementation and Changeover (5%, K2)

The objective is to recognise the importance of careful implementation.

11.1 The task of file and data conversion
- Technical feasibility of converting data
- Alternatives to automatically converting data, such as printing data out and re-entering it

11.2 The principles and problems of data mapping
- The principles of data mapping
- Common problems in data mapping, such as field type incompatibility, field length differences, different field structures and absence of required fields in the current system
- Approaches to dealing with different field structures (for example, converting from one address field to three address line fields) and issues arising from populating newly defined fields with valid data

11.3 Plan, test and undertake data conversion
- Plan the steps and the timing of data conversion
- Write and test the data conversion programs
- The possible role of an automated test comparator in the testing process
- Undertake the actual conversion of live data

11.4 The role of supporting documentation, including user manuals
The role and structure of an online help facility
• The role and structure of a printed user guide or user manual
• The role and structure of printed operational manuals
• The role and structure of technical documentation designed to allow the continuing support of the delivered software

11.5 Approaches to training
• Conventional lectures and workshops
• Remote mechanisms, such as web-casts and tele-conferencing
• Computer-based training (CBT) and e-learning initiatives

11.6 Define training needs and evaluate training effectiveness
• Identify current and proposed competencies
• Define an appropriate strategy, using approaches to training (11.5), to support the gaining of proposed competencies
• Assess the effectiveness of training in supporting these proposed competencies through post-course questionnaires and tests

11.7 Systems implementation
• The principles of direct changeover/conversion
• The advantages and disadvantages of direct changeover/conversion within a given scenario
• The principles of parallel running
• The advantages and disadvantages of parallel running within a given scenario
• The principles of pilot running
• The advantages and disadvantages of pilot running within a given scenario

12. Evaluation and Maintenance (5%, K2)
The objective is to recognise the need to evaluate a delivered system and to enhance it through subsequent maintenance.

12.1 The location of maintenance in the systems development life cycle
• Maintenance in abbreviated waterfall and V models
• The explicit reference to maintenance in the b model
• Maintenance in an iterative environment

12.2 The range of metrics which might be used to evaluate a delivered software product
• Characteristics of good metrics (for example, quantifiable, relevant, easy to collect)
• Metrics associated with the business objectives of the project (for example, concerned with improving profitability of the organisation)
• Metrics associated with the functional fit of the delivered software product (for example, number of corrective maintenance changes raised after implementation)
• Metrics associated with the reliability of the software product (for example, acceptable downtime expressed as a percentage)
• Metrics associated with the usability of the software product (for example, time taken to learn the product to an acceptable level)

12.3 The purpose and conduct of a post-implementation review
• The purpose of the post-implementation review is primarily to examine the success of the software product against pre-defined criteria (12.2) and to discuss user feedback and outstanding issues
• Participants in the post-implementation review
• Outcomes from the post-implementation review concerning the delivered software product

12.4 The purpose and conduct of a post-project review
• The purpose of the post-project review is primarily to examine the conduct of the project that delivered the software product. Focus is on what went badly and well with the aim of implementing lessons learnt into improved project or systems development processes
• Participants in the post-project review
• Outcomes from the post-project review concerning the way projects or systems development is conducted

12.5 The distinction between corrective, adaptive and perfective maintenance
• Corrective maintenance is concerned with fixing software faults and failures. This includes fixing requirements which were incorrectly defined and implemented in the previous (or initial) software product release
• Adaptive maintenance is concerned with implementing new or changed requirements into the software product. For example, implementing changes in payroll rules that have arisen since the previous (or initial) software product release
• Perfective maintenance is concerned with making changes to the software to improve, for example, its usability or maintainability. These changes are not as a result of faults, failures or changes. They are improvements to the general product quality of the software release

13. Software Support Tools (5%, K2)

The objective is to appreciate the wide range of software support tools, which support IT development.

13.1 Computer Aided Software Engineering tools (CASE)
• The objectives of CASE tools
• Relationship of the CASE tools elements to the software development life cycle
• Facilities of CASE tools
• Advantages and disadvantages of CASE tools
13.2 Computer Aided Software Testing tools (CAST)
  - The objectives of CAST tools
  - Relationship of the CAST tools to the software testing life cycle
  - Facilities of CAST tools
  - Advantages and disadvantages of CAST tools

13.3 Configuration Management tools (CM)
  - The objectives of CM tools
  - The importance of CM within the software life cycle
  - Facilities of CM tools
  - Advantages and disadvantages of CM tools
Recommended Reading

Title: Developing Information Systems: Practical Guidance for IT Professionals  
Author: James Cadle (editor)  
Publisher: BCS  
Publication Date: August 2014  
ISBN: 978-1780172453  
URL: http://shop.bcs.org
Levels of Knowledge / SFIA Levels

This course will provide candidates with the levels of difficulty / knowledge skill highlighted within the following table, enabling them to develop the skills to operate at the levels of responsibility indicated.

The levels of knowledge and SFIA levels are explained in on the website www.bcs.org/levels.

The levels of knowledge will also enable candidates to develop the following levels of skill to be able to operate at the following levels of responsibility (as defined within the SFIA framework) within their workplace:

<table>
<thead>
<tr>
<th>Level</th>
<th>Levels of Knowledge</th>
<th>Levels of Skill and Responsibility (SFIA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K7</td>
<td>Set strategy, inspire and mobilise</td>
<td></td>
</tr>
<tr>
<td>K6</td>
<td>Evaluate</td>
<td>Initiate and influence</td>
</tr>
<tr>
<td>K5</td>
<td>Synthesise</td>
<td>Ensure and advise</td>
</tr>
<tr>
<td>K4</td>
<td>Analyse</td>
<td>Enable</td>
</tr>
<tr>
<td>K3</td>
<td>Apply</td>
<td>Apply</td>
</tr>
<tr>
<td>K2</td>
<td>Understand</td>
<td>Assist</td>
</tr>
<tr>
<td>K1</td>
<td>Remember</td>
<td>Follow</td>
</tr>
</tbody>
</table>

Question Weighting

<table>
<thead>
<tr>
<th>Syllabus Area</th>
<th>Target number of questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction to Systems Development</td>
<td>2</td>
</tr>
<tr>
<td>2. Lifecycle types and rationales</td>
<td>4</td>
</tr>
<tr>
<td>3. Business Analysis</td>
<td>2</td>
</tr>
<tr>
<td>4. Requirements Engineering</td>
<td>4</td>
</tr>
<tr>
<td>5. Making a Business Case</td>
<td>4</td>
</tr>
<tr>
<td>6. Programming and Development</td>
<td>4</td>
</tr>
<tr>
<td>7. Systems Modelling and Specification</td>
<td>4</td>
</tr>
<tr>
<td>8. Systems Design</td>
<td>4</td>
</tr>
<tr>
<td>9. System Architecture</td>
<td>2</td>
</tr>
<tr>
<td>10. Quality and Testing</td>
<td>4</td>
</tr>
<tr>
<td>11. Implementation and Changeover</td>
<td>2</td>
</tr>
<tr>
<td>12. Evaluation and Maintenance</td>
<td>2</td>
</tr>
<tr>
<td>13. Software Support Tools</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>40 Questions</strong></td>
</tr>
</tbody>
</table>
### Format of Examination

<table>
<thead>
<tr>
<th>Type</th>
<th>40 Multiple Choice Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>60 minutes. Candidates are entitled to an additional 15 minutes if they are sitting an examination in a language that is not their native/official language.</td>
</tr>
<tr>
<td>Pre-requisites</td>
<td>Accredited training is strongly recommended but is not a pre-requisite.</td>
</tr>
<tr>
<td>Supervised</td>
<td>Yes</td>
</tr>
<tr>
<td>Open Book</td>
<td>No (No reading materials allowed into the examination room)</td>
</tr>
<tr>
<td>Pass Mark</td>
<td>26/40 (65%)</td>
</tr>
<tr>
<td>Calculators</td>
<td>Calculators are not allowed during this examination</td>
</tr>
<tr>
<td>Learning Hours</td>
<td>21 Hours</td>
</tr>
<tr>
<td>Delivery</td>
<td>Paper based examination via a BCS Accredited Training Organisation</td>
</tr>
</tbody>
</table>

### Trainer Criteria

<table>
<thead>
<tr>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Hold the BCS Foundation Certificate in Systems Development</td>
</tr>
<tr>
<td>- Have a minimum of 3 years’ practical system/solution development experience, a minimum of 2 years’ training experience or 1 year with a recognised training qualification</td>
</tr>
<tr>
<td>- Have a minimum of 10 days’ training experience or hold a train the trainer qualification</td>
</tr>
</tbody>
</table>

### Classroom Size

<table>
<thead>
<tr>
<th>Trainer to candidate ratio</th>
<th>1:16</th>
</tr>
</thead>
</table>