BCS Level 3 Certificate in Software Development Context and Methodologies Syllabus QAN 603/1191/5

Version 3.2
July 2020

This is a United Kingdom government regulated qualification which is administered and approved by one or more of the following: Ofqual, Qualification in Wales, CCEA Regulation or SQA

BCS Level 3 Certificate in Software Development Context and Methodologies

Contents

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.

Version Number	Changes Made
Version 1.0 March 2016	Syllabus Created
Version 1.1 October 2016	Amendment made.
Version 1.2 December 2016	Compliance statement added.
Version 2.0 August 2017	Major amendments following full review
Version 2.1 October 2017	Minor amendments following review
Version 2.2 October 2017	Minor amendments following review
Version 3.0 October 2017	Minor amendment
Version 3.1 March 2018	Watermark removed
Version 3.2 July 2020	Syllabus amendments to make it suitable for a range of different learners as well as apprentices

Introduction

This Certificate is the first module of the two knowledge modules required for the Level 3 Software Development Technician apprenticeship programme. It covers the range of concepts, approaches and techniques that are applicable to Software Development Technicians, for which learners are required to demonstrate their knowledge and understanding.

This certificate can also be delivered as a standalone programme for learners working to develop their knowledge and understanding.

Objectives

Learners should be able to demonstrate knowledge, understanding and the underlying principles and processes of the following key areas:

1. Appreciate the business context and marketing environment for software development.
2. Recognise that there are different methodologies that can be used for software development.
3. Understand roles within the software development team.
4. Understand the structure of software applications and the particular context for multiple development platforms.
5. Appreciate all stages of the software development lifecycle.
6. How software testing contributes in the production of software and systems to a known quality.
7. Understand the role of configuration management and version control systems and how to apply them. Understanding and complying with business processes.
8. Working knowledge of business IT skills relevant to the organisation.

Learners who are completing this as part of the apprenticeship programme should collate evidence of lessons learnt in these key areas and these should be reflected upon when the learner is compiling the Summative Portfolio. This will provide the learner with the opportunity to identify how the task might be done better/differently with knowledge subsequently gained.

Target Audience

The certificate is relevant to learners who are either enrolled on the Level 3 Software Development Technician apprenticeship programme or want to gain more understanding of software development to a level 3 standard.

Course Format and Duration

Candidates can study for this certificate by attending a training course provided by a BCS accredited training provider. The estimated total qualification time for this award is 125 hours.

Eligibility for the Examination

Apprenticeship learners:
Individual employers will set the selection criteria, but this is likely to include 5 GCSEs (especially English, mathematics and a science or technology subject); other relevant qualifications and experience; or an aptitude test with a focus on IT skills. Learners should have the ability to solve logical problems and have experience in data and code writing. Level 2 English and Maths will need to be achieved, if not already, prior to taking the endpoint assessment.

Other learners:
It is recommended that learners have completed 5 GCSEs (especially English, mathematics and a science or technology subject); other relevant qualifications and experience; or an aptitude test with a focus on IT skills. Learners should have the ability to solve logical problems and have experience in data and code writing.

Format and Duration of the Examination

The format for the examination is a 60-minute multiple-choice examination consisting of 40 questions. The examination is closed book (no materials can be taken into the examination room). The pass mark is 26/40 (65%).

Additional Time for Learners Requiring Reasonable Adjustments Due to a Disability

Learners may request additional time if they require reasonable adjustments. Please refer to the reasonable adjustments policy for detailed information on how and when to apply.

Additional Time for Learners Whose Language is Not the Language of the Examination

If the examination is taken in a language that is not the learner's native / official language, then they are entitled to 25% extra time.

If the examination is taken in a language that is not the learner's native / official language, then they are entitled to 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.

Guidelines for 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. Training providers may spend more time than is indicated and learners 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.

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. Learners enrolled on an apprenticeship programme should be encouraged to consider their summative portfolio throughout the modules.

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 Business context and market environment (13%, K2)

Understand the business context and market environment for software development.

1.1 Understand how similar software development processes and methods are used across a range of industries but can be based on very different rationale.

- data.

1.2 Identify the factors that may lead to the development of different information systems within or across a range of industry sectors, including.

- business requirements;
- project timescales;
- budgets;
- resources and skills availability;
- product and project risks.

1.3 Explain why businesses need to keep digital processes up to date and web systems responsive to user needs.

1.4 Explain the difference between virtual web-based enterprises and companies that use web and digital services with respect to customer and client engagement.

2 Software Development Methodologies (13%, K2)

Recognise that there are different methodologies that can be used for software development.

2.1 Identify the main features of sequential development methods and approaches.

- Waterfall.

2.2 Identify the main features of iterative (incremental) development methods and approaches.

- Agile.

2.3 Distinguish between the use of different software development methodologies and approaches, considering their suitability and application to the project.

- Agile;
- Waterfall.

3 Team Roles and Relationships (14%, K2)

Understands the roles within the software development team.

3.1 Describe the main roles within software development teams.

- requirements engineer;
- business analyst;
- software designer;
- software developer;
- software tester;
- software project manager;
- software release engineer.

3.2 Distinguish how the different roles (as listed in 3.1) relate / work with each other and their key accountabilities, in order to complete specific activities and tasks.

3.3 Recognise the key external roles and processes that interface to the roles within the software development team (as listed in 3.1).

- customers;
- end-users;
- operation’s processes and personnel;
- service management processes and personnel.

3.4 Recognise that collaborative approaches are especially important in Agile development and DevOps practices.

4 Application Structure and Development Platform Context (15%, K2)

Understand the structure of software applications and the particular context for the development platform (whether web, mobile, or desktop applications).

4.1 Identify the different components that contribute to the underlying architecture of software applications.

- code and libraries;
- data;
- application components;
- application interfaces;
    - network and hardware platforms
- reference to the OSI (Open Systems Interconnection) model.

4.2 Describe the use of data sources in software applications for storage and retrieval of information.

4.3 Explain the features of the following platforms in context of software development, deployment and underlying architecture.

- web;
- desktop;
- mobile;
- server;
- cloud.

4.4 Distinguish the characteristics of software development that are impacted by the deployment of software.

- on multiple platforms;
- to a single platform.

5 The Software Development Lifecycle (SDLC) (14%, K2)

Understands all stages of the software development lifecycle.

5.1 Recognise that there are several ways to represent the terminology and phases of the SDLC.

- feasibility study;
- requirements analysis;
- design;
- code development;
- testing;
- deployment / implementation;
- maintenance.

5.2 Summarise the phases of the SDLC.

5.3 Identify the main activities of each of the phases of the SDLC in terms of inputs, activities and outputs.

5.4 Recognise the relationship between the phases of the SDLC and the roles within the software development team.

6 Software Testing (19%, K2)

Understand how to test code (e.g. unit testing).

6.1 Recognise why testing is necessary, including principles of:

- early testing;
- risk reduction;
- conformance to functional and non-functional requirements;
- finding and reporting defects;
- the difference between testing and debugging.

6.2 Summarise the different levels of testing within the SDLC.

- unit;
- integration;
- system;
- acceptance.

6.3 Describe how unit testing follows the fundamental test process consisting of:

- test planning, monitoring and control, including maintaining traceability between requirements and testing artefacts;
- test analysis and design;
- test implementation and execution;
- evaluating exit criteria and reporting.

6.4 Identify the different types and techniques for software testing that are available and why they would be used, including:

- functional testing;
- non-functional testing;
    - security
    - performance
    - reliability
- reviews and static analysis;
- white box testing (structure-based);
- black box testing (specification-based).

6.5 Recognise the tool types used to support software testing and their main purpose.

- test management;
- static testing;
- test execution;
- performance / load / stress testing.

7 Configuration Management and Version Control Systems (12%, K2)

Understand the role of configuration management and version control systems and how to apply them.

7.1 Explain how configuration management tools and techniques are used to control and manage the different software development artefacts through the phases of the SDLC and live operation, including:

- requirements documentation;
- code;
- test scripts.

7.2 Summarise the main features and benefits of version control for the development of code including:

- change history;
- concurrent working;
- tracking and preventing conflicts;
- traceability;
- security.

7.3 Explain how version control can be used for software and software artefacts.

- that are being developed for use on multiple platforms;
- where similar but slightly different versions need to be produced.

Levels of Knowledge / SFIA Levels

This syllabus will provide learners 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 on the website www.bcs.org/levels. The levels of knowledge above will enable learners 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:

Level	Levels of Knowledge	Levels of Skill and Responsibility (SFIA)
K7		Set strategy, inspire and mobilise
K6	Evaluate	Initiate and influence
K5	Synthesise	Ensure and advise
K4	Analyse	Enable
K3	Apply	Apply
K2	Understand	Assist
K1	Remember	Follow

Question Weighting

Syllabus Area	Target number of questions
1. Business Context and Market Environment	5
2. Software Development Methodologies	5
3. Team Roles and Responsibilities	6
4. Application and Development Platform Context	6
5. The Software Development Lifecycle (SDLC)	5
6. Software Testing	8
7. Configuration Management and Version Control Systems	5
Total	**40 Questions**

Format of Examination

Type	40 Question Multiple Choice.
Duration	60 minutes. An additional 25% will be allowed for learners sitting the examination in a language that is not their native / mother tongue.
Pre-requisites	Training from a BCS Accredited Training Provider is strongly recommended but is not a pre-requisite.
Supervised	Yes
Open Book	No
Pass Mark	26/40 (65%).
Calculators	Calculators cannot be used during this examination.
Total Qualification Time (TQT)	125 Hours.
Delivery	Online

Trainer Criteria

Criteria	▪ Have 10 days training experience or have a train the trainer qualification ▪ Have a minimum of 3 years practical experience in the subject area

Classroom Size

Trainer to learner ratio	1:16