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

Version 1.0
March 2017

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	Final Draft created
Version 0.1 October 2016	
Version October 2016	Document created

Introduction

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

Objectives

Overall, apprentices should be able to demonstrate knowledge and understanding of software development and its underlying architecture, it's principles and techniques. This module focuses on software development context and methodologies, with the key topics as follows:

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.

Evidence of lessons learnt in these key areas should be collected and reflected upon when the apprentice is compiling the Summative Portfolio. The apprentice should identify how the task might be carried out better or differently with knowledge subsequently gained.

Target Audience

The Certificate is relevant to anyone enrolled on the Level 3 Software Development Technician Apprenticeship Programme.

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 170 hours.

Eligibility for the Examination

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.
Level 2 English and maths will need to be achieved, if not already, prior to taking the end point assessment.

Format and Duration of the Examination

The format for the examination is a one-hour 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 Apprentices Requiring Reasonable Adjustments Due To a Disability

Apprentices 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 Apprentices Whose Language Is Not the Language of the Exam

If the examination is taken in a language that is not the apprentice's native/official language, then they are entitled to 25% extra time.
If the examination is taken in a language that is not the apprentice'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 apprentices 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. Apprentices 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 shown relates to the proportion of the exam given to that area.
The K level identifies the maximum knowledge level that may be tested for that area.

1. Business Context and Market Environment (10%, K2)

In this topic, the apprentice will learn that a broad range of industries and markets differ in their approach to information and their rationale, whilst the software development processes and methods are similar. The successful apprentice should be able to:

1.1 Recognise that business, market and information needs may lead to very different applications whilst using similar development processes and methods.

1.2 Explain the different business and market contexts that lead to varying approaches to software development, including but not limited to:

- Characteristics of market sectors and segments
- Business drivers and desired outcomes
- Market risk
- Operational risk
- Legal and regulatory requirements

1.3 Explain the reasons why businesses need to keep processes and applications up to date, and responsive to user and customer needs.

1.4 Distinguish types of business in context of their provision and use of web and digital services.

2. Software Development Methodologies (20%, K3)

In this topic, the apprentice will learn a broad range of methodologies, frameworks and how they can be applied to software development. The successful apprentice should be able to:

2.1 Recognise and list a variety of software development methodologies which can be broadly categorised as sequential or iterative/incremental and that there are a variety of specific methods/approaches within them.

2.2 Summarise the main differences between sequential and iterative/incremental methodologies.

2.3 Explain the main principles of Agile together with the different approaches within it including but not limited to:

- Scrum
- Kanban
- Extreme programming (XP)
- Test Driven development (TDD)
- Behavior driven development (BDD)

2.4 Demonstrate how software development is undertaken within governance and management frameworks/methods that organisations use to control and manage projects, programmes and operations.

3. Team Roles and Relationships (10%, K2)

In this topic, the apprentice will learn the roles involved in software development and how they relate to each other within a software development team. The successful apprentice should be able to:

3.1 Describe the main roles within software development teams, including but not limited to:

- Requirements engineer
- Business analyst
- Software designer
- Software developer
- Software tester
- Software project manager
- Software release engineer

3.2 Explain how the different roles relate to each other in terms of typical interfaces and accountabilities.

3.3 Recognise the key roles and processes external to the software development team which interface with the software development team, including but not limited to:

- Customers
- Users
- Operations
- Service management

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

4. Application Structure and Development Platform Context (20%, K3)

In this topic, the apprentice will learn and appreciate the underlying architecture of software applications, and the importance of linking software to databases for data storage and information presentation to users. The apprentice will recognise the different requirements for development and deployment of software across multiple platform types. The successful apprentice should be able to:

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

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, including but not limited to:

- Web
- Mobile
- Desktop
- Server
- Cloud

4.4 Distinguish the characteristics of software development that are impacted by the deployment of software on multiple platforms as opposed to a single platform.

5. The Software Development Lifecycle (SDLC) (10%, K3)

In this topic, the apprentice will learn and gain an appreciation of how software development follows a structured and phased approach to different activities through the software development lifecycle. The successful apprentice should be able to:

5.1 Recognise that there are several ways to represent the terminology and phases of the SDLC (Software development lifecycle) rationalised here as follows:

- Feasibility Study
- Requirements Analysis
- Design
- Code Development
- Testing
- Deployment/Implementation
- Maintenance

5.2 Summarise the phases of the SDLC. (Software development lifecycle)

5.3 Explain the main activities of each of the phases of the SDLC (Software development lifecycle) in terms of inputs, activities and outputs.

5.4 Show how the phases of the SDLC (Software development lifecycle) may be implemented according to the development methodology and approach adopted for a project or organisation.

6. Software Testing (20%, K3)

In this topic, the apprentice will learn and understand the levels and types of software testing, their purpose and examples of their application. The successful apprentice should be able to:

6.1 Explain why testing is necessary, including but not limited to principles of:

- Early testing
- Risk reduction
- Conformance to functional and non-functional requirements
- Finding defects
- Differentiating testing from debugging

6.2 Within the SDLC (Software development lifecycle) summarise the different levels of testing, their purposes and the generic test process that supports them all:

- Unit
- Integration
- System
- Acceptance

The generic test process, activities and key deliverables, including but not limited to:

- Test analysis, design, implementation, execution, reporting
- Test planning, monitoring & control
- Test conditions, test cases, test scripts
- Information traceability between tests and requirements

6.3 Describe different types and techniques for software testing that are available and why they would be used, including but not limited to:

- Functional Testing
- Non-Functional Testing
  - Security
  - Performance
  - Usability
  - Reliability
- Regression testing
- Reviews and static analysis
- White Box testing (Structure-based)
- Black Box testing (Specification-based)
  - Equivalence partitioning
  - Boundary value analysis
  - Decision tables
  - State transition diagrams
- Experience-based testing

6.4 Write test cases including coverage measurement from given requirements (text or model) using the following Black Box (Specification-based) techniques:

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition diagrams/tables

6.5 For tools to support software testing successful apprentices should be able to:

- Recognise the tool types to support software testing and their main purpose
- Summarise the pros and cons of test automation

7. Configuration Management and Version Control Systems (10%, K3)

In this topic, the apprentice will gain an understanding of the importance of configuration management and version control, together with the use of tools and techniques to store and manage incremental updates and versions of code. The successful apprentice should be able to:

7.1 Explain how configuration management tools and techniques are used to control and manage the different software development artifacts (including but not limited to) through the phases of the SDLC (Software development lifecycle) and live operation.

- Requirements documentation
- Code
- Test scripts

7.2 Summarise the main features and benefits of version control for the development of code, including but not limited to:

- Change history
- Concurrent working
- Tracking and preventing conflicts
- Traceability
- Security

7.3 Demonstrate how version control, in conjunction with configuration and release management, can be used for software that is being developed for use on multiple platforms.

7.4 Discuss different version control approaches and their relative pros and cons.

Levels of Knowledge / SFIA Levels

This syllabus will provide apprentices 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 apprentices 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	4
2.	Software Development Methodologies	8
3.	Team Roles and Relationships	4
4.	Application Structure and Development Platform Context	8
5.	The Software Development Lifecycle	4
6.	Software Testing	8
7.	Configuration Management and Version Control Systems	4
Total		**40 Questions**

Format of Examination

Type	40 Question Multiple Choice.
Duration	1 Hour. An additional 15 minutes will be allowed for apprentices 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)	170 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 Apprentice ratio	1:16

Recommended Reading List

Title	Achieving Software Quality Through Teamwork
Author	Isabel Evans
Publisher	Artech House
Publication Date	30 April 2004
ISBN	978-1580536622
URL	https://www.amazon.com/Achieving-Software-Quality-through-Teamwork/dp/158053662X

Title	Software Testing: An ISTQB-BCS Certified Tester Foundation Guide 3rd edition
Author	Angelina Samaroo, Geoff Thompson, Brian Hambling
Publisher	BCS
Publication Date	17 June 2015
ISBN	978-1780172996
URL	https://www.amazon.co.uk/d/Books/Software-Testing-ISTQB-BCS-Certified-Tester-Foundation-Guide/1780172990

Title	ITIL and the Software Lifecycle: Practical Strategy and Design Principles
Author	Brian Johnson and John Higgins
Publisher	Van Haren Publishing
Publication Date	August 2007
ISBN	978-90875304495
URL	https://www.amazon.com/ITIL-Software-Lifecycle-Principles-Management/dp/9087530498

Title	Configuration Management Best Practices: Practical Methods that work in the Real World
Author	Bob Aleilo, Leslie Sachs
Publisher	Addison-Wesley
Publication Date	10 August 2010
ISBN	978-0321685865
URL	**https://www.amazon.co.uk/Configuration-Management-Best-Practices-Practical/dp/0321685865**

Title	Software Configuration Management patterns: Effective Teamwork, Practical Integration
Author	Stephen P Berczuk, Brad Appleton
Publisher	Addison-Wesley
Publication Date	4 Nov 2002
ISBN	978-0201741179
URL	**https://www.amazon.co.uk/Software-Configuration-Management-Patterns-Integration/dp/0201741172**

Title	Specification by Example: How Successful Teams deliver the Right Software
Author	Gojko Adzic
Publisher	Manning
Publication Date	6 June 2011
ISBN	978-1617290084
URL	**https://www.amazon.co.uk/Specification-Example-Successful-Deliver-Software/dp/1617290084**

Title	Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation
Author	Jez Humble, David Farley
Publisher	Addison-Wesley
Publication Date	27 July 2010
ISBN	978-0321601919
URL	**https://www.amazon.co.uk/d/Books/Continuous-Delivery-Deployment-Automation-Addison-Wesley/0321601912**

Title	Software Configuration Management Handbook – 3rd Edn
Author	Alexis Leon
Publisher	Artech House
Publication Date	30 Nov 2004
ISBN	978-1608078431
URL	**https://www.amazon.com/Software-Configuration-Management-Handbook-Alexis/dp/1608078434**

Title	Test-Driven Development: By Example
Author	Kent Beck
Publisher	Addison-Wesley
Publication Date	8 Nov 2002
ISBN	978 0321146533
URL	https://www.amazon.co.uk/d/cka/Test-Driven-Development-Addison-Wesley-Signature-Kent-Beck/0321146530

Title	Agile Software Development, Principles, Patterns, and Practices, 01 Edition
Author	Robert C Martin
Publisher	Pearson
Publication Date	17 July 2013
ISBN	978-1292925940
URL	

Title	Scrum – A Pocket Guide
Author	Verheyen, Gunter
Publisher	Van Haren Publishing
Publication Date	2013
ISBN	978-9087537203
URL	https://www.amazon.co.uk/Scrum-Practice-Publishing-Verheyen-Paperback/dp/B00ZVO7R0Q

Relevant Standards:

ISO/IEC 12207:2008 - Systems and Software Engineering - software lifecycle processes
https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en

https://webstore.iec.ch/preview/info_isoiec12207%7Bed2.0%7Den.pdf

ISO/IEC/IEEE 15288:2015 - System and Software Engineering - System lifecycle processes
http://www.iso.org/iso/catalogue_detail?csnumber=63711

ISO/IEC/IEEE 24765: 2010 - System and Software Engineering Vocabulary
http://www.iso.org/iso/catalogue_detail.htm?csnumber=50518

IEEE 828-2012: IEEE Standard for Configuration Management in Systems and Software Engineering (which supports the above ISO standards)
https://standards.ieee.org/findstds/standard/828-2012.html

ISO/IEC/IEEE 29119:2013 – Software and Systems Engineering - Software Testing
http://www.iso.org/iso/catalogue_detail.htm?csnumber=45142

ISO/IEC 25010:2011 – Systems and software Quality Requirements and Evaluation (SQuaRE)
http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733

Useful Websites

- https://www.usability.gov/
- https://www.w3.org/
- https://www.microsoft.com/en-us/sdl/