**Digital Apprenticeships at Gloucestershire College**

**Software Developer**

****

Contents

[Introduction 3](#_Toc501008563)

[Qualification 3](#_Toc501008564)

[Entry Requirements 3](#_Toc501008565)

[Duration 3](#_Toc501008566)

[What you will study at college 3](#_Toc501008567)

[In the Workplace 4](#_Toc501008568)

[Grading the summative portfolio 5](#_Toc501008569)

[Progress Monitoring 5](#_Toc501008570)

[Training plan 5](#_Toc501008571)

[Appendix 1: Software Developer Skills Audit 7](#_Toc501008572)

[Appendix 2: Competency standards 8](#_Toc501008573)

[The what – what the apprentice has shown they can do 8](#_Toc501008574)

[The how: the way in which the work has been done 12](#_Toc501008575)

[The with whom: the personal and interpersonal qualities the apprentice has brought to internal and external relationships 14](#_Toc501008576)

## Introduction

This handbook contains specific information about the Software Developer apprenticeship.

See the general handbook for information common to all the digital apprenticeships.

The primary role of a software developer is to build and test simple, high-quality code across front end, logic and database layers. A developer will typically be working as part of a larger team, in which they will have responsibility for some of the straightforward elements of the overall project. The developer will need to be able to interpret design documentation and specifications. The customer requirements will typically be defined and agreed by more experienced or specialist members of the team, such as a business analyst or technical architect.

Job roles include:

* Web Developer
* Application Developer
* Mobile App Developer
* Games Developer
* Software Developer

## Qualification

Level 4 Software Developer

## Entry Requirements

Individual employers will set the selection criteria, but this is likely to include 2 A levels in STEM subjects **OR** a level 3 apprenticeship or other relevant qualification **OR** relevant experience and/or an aptitude test with a focus on functional maths.

## Duration

A typical apprenticeship will last between 18 and 24 months including an end point assessment (EPA).

It consists of 80% on the job training and 20% of off the job training.

|  |  |
| --- | --- |
| **On the job training and projects** | 80% |
| **College Day Release** | 10% |
| **Assignments/Portfolio** | 10% |
| **TOTAL** | 100% |

## What you will study at college

There are two BCS knowledge modules studied in the following order:

|  |
| --- |
| Software Development Methodologies (for Level 4 Software Development Apprenticeship) |
| Software Languages (for Level 4 Software Developer Apprenticeship) |

You will be required to complete a module before commencing the next module. Each module has a vendor/professional qualification which needs to be completed by the apprentice. The modules are made up of:

* Technical Competencies
* Technical knowledge and Understanding
* Underpinning Skills, Attitudes and Behaviours

The BCS exams have a pass mark of 65%.

External tests will come in the form of automated tests. The tests offer instant results to the learner. We will provide sample tests on an ongoing basis.

All the topics for each module are shown in Appendix 1.

## In the Workplace

You will be set projects by your employer and your assessor. You will use the programming languages appropriate to your job. These will allow you to show what you can do, the way in which you have done it and the relationships with your colleagues. You will build a portfolio of evidence leading to a final (summative) portfolio for external assessment.

All evidence for the portfolio will be submitted electronically to the college’s electronic assessment software (accessed by web browser)

The purpose of the portfolio is to demonstrate that you have attained all the knowledge and skills shown in this table:

|  |  |
| --- | --- |
| **Qualification Level 4 Descriptor** | |
| Knowledge descriptor (the holder…) | Has factual, procedural and theoretical knowledge and understanding of a subject or field of work to complete tasks and address problems that while well-defined, may be complex and non-routine. Can interpret and evaluate relevant information and ideas. Is aware of the nature of the area of study or work. Is aware of different perspectives or approaches within the area of study or work. |
| Skills descriptor (the holder should have…) | * Logical and creative thinking skills * Analytical and problem solving skills * Ability to work independently and to take responsibility * Own initiative * A thorough and organised approach * Ability to work with a range of internal and external people * Ability to communicate effectively in a variety of situations * Maintain productive, professional and secure working environment. |

Competency in the workplace, as evidenced by the portfolio, is divided into what you have done, the way in which you did it and with whom you worked. You need to meet the standards to gain a pass, but if you significantly exceed the standards you can be awarded a merit or a distinction. This is decided by the external assessor during end point assessment.

Details of the competency standards assessed in the portfolio are shown in appendix 2.

## Grading the summative portfolio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GRADE | The WHAT | | The HOW | | With WHOM | |
| What you have shown you can do | | The way in which you have done the work | | The personal and interpersonal qualities you brought to work relationships | |
| Met expectations | Significantly exceeded expectations | Met expectations | Significantly exceeded expectations | Met expectations | Significantly exceeded expectations |
| DISTINCTION |  |  |  |  |  |  |
| MERIT |  |  |  |  |  |  |
| MERIT |  |  |  |  |  |  |
| PASS |  |  |  |  |  |  |

## Progress Monitoring

You will regularly be visited by your appointed assessor (normally monthly). Both you and your line manager will be required so that new projects can be agreed

There will be a review of

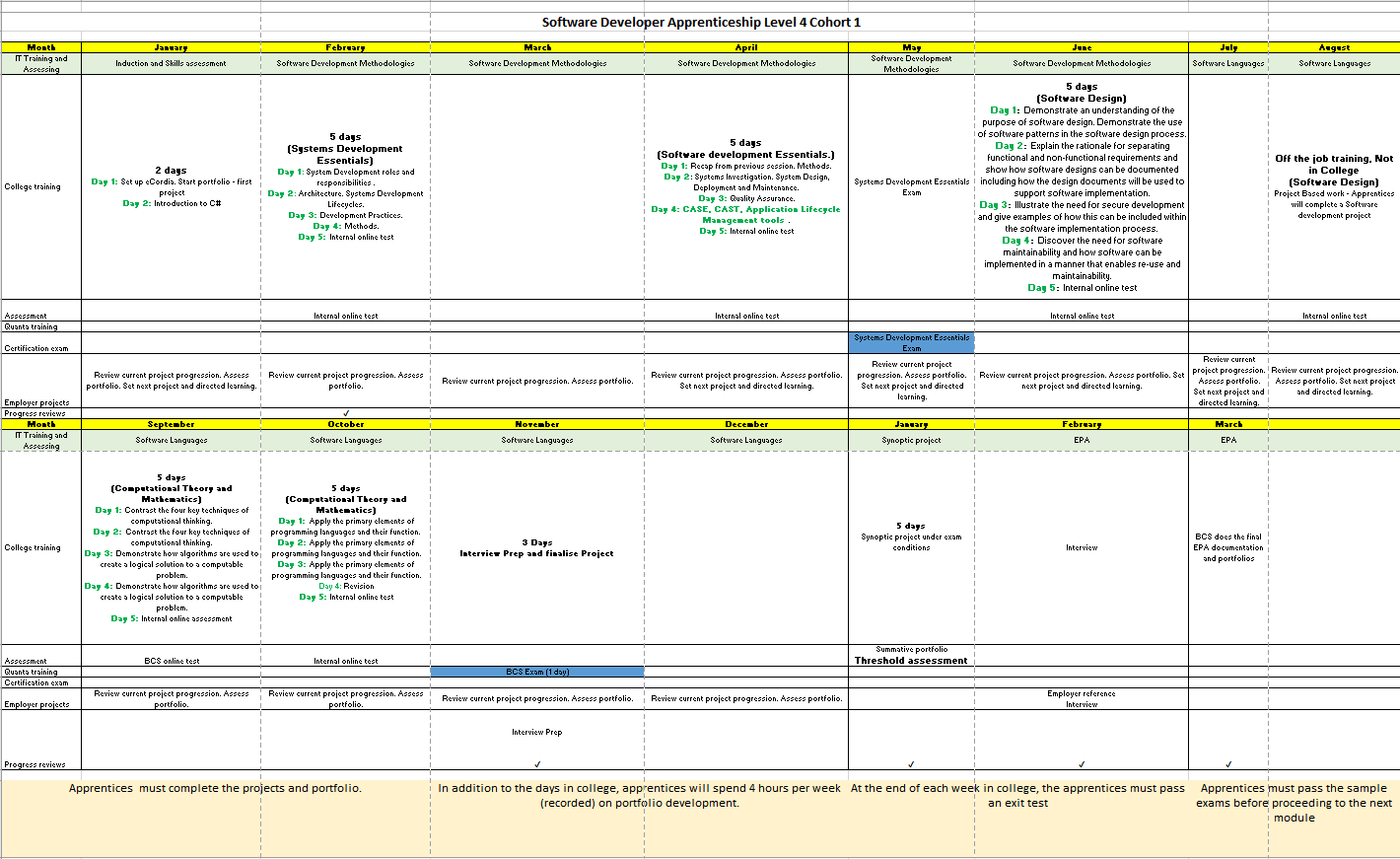
* your portfolio
* the record of assignment and portfolio time spent away from normal duties
* your progress against expectation
* the projects and tasks to be completed before the next visit
* the date and time of the next visit

All progress will be monitored via the college’s electronic assessment software (accessed by web browser)

*The training coordinator will visit you regularly. At this meeting she will evaluate your overall performance, attendance, progress, etc.*

## Training plan

The table on the next page shows all the activities for the apprenticeship by month.



# Appendix 1: Software Developer Skills Audit

The items in the columns are from the BCS Knowledge Module (KM). They cover all the topics you are required to know.

Tick the box in each column that most matches your knowledge and competence for that item.

means you do not know this topic

0 means you have some knowledge of the topic

means you are confident about this topic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Standard | Minimum requirement |  | 0 |  |
| Understands and operates at all stages of the software development lifecycle (KM1) | a) What is the software development lifecycle (SDLC) |  |  |  |
| b) What are the seven generic stages and their high- level deliverables from each stage |  |  |  |
| c) What are the main activities in each stage |  |  |  |
| Understands the similarities and differences (taking into account positives and negatives of both approaches) between agile and waterfall software development methodologies (KM1) | a) What is the agile development method |  |  |  |
| b) What is the waterfall development method |  |  |  |
| c) What are the strengths and weaknesses of both approaches |  |  |  |
| Understands how teams work effectively to produce software and contributes appropriately (KM1) | a) What are the roles that need to be undertaken b) How are these roles distributed across a team c) What team-working aspects are needed to ensure effective delivery of projects |  |  |  |
| Understands and applies software design approaches and patterns and can interpret and implement a given design, compliant with security and maintainability requirements (KM2) | a) Software design approaches  b) Software patterns c) Documenting software designs d) Secure development e) Designing for software maintainability and re-use |  |  |  |
| Understands and applies the maths required to be a software developer (e.g. algorithms, logic and data structures) (KM2) | Understanding of basic algorithmic processing to define the problem and/or solution  Elements of programming logic - variables; assignment statements; data types; conditionals; loops; arrays; and input/output,  Knowledge of at least two data structures – such as Arrays or Collection Classes |  |  |  |

# Appendix 2: Competency standards

## The what – what the apprentice has shown they can do

|  |  |
| --- | --- |
| Competency Standard | Minimum expected requirements |
| Logic: writes good quality code (logic) with sound syntax in at least one language | Apprentices can write code to achieve the desired functionality and which is easy to read and understand, with good naming, indentation and commenting, and applying the fundamentals of good coding:   * development paradigms (where this is object oriented programming this must include inheritance, abstractions, encapsulation, polymorphism) * software programming languages * software development tools (IDEs) * writing programs and methods * language-specific idioms * logic and flow-of-control   Apprentices can apply:   * Elements of programming – variables, assignment statements, data types, conditionals, loops, arrays, and input/output. * Functions - modular programming dividing a program into components that can be independently debugged, maintained, and reused writing at least two reusable functions * Algorithms and data structures - classical algorithms for sorting and searching, and fundamental data structures. |
| User interface: can develop effective user interfaces for at least one channel | Apprentices can apply the fundamental concepts of human-computer interaction or user experience design, the development practices leading to a high-quality user interface, and the programming techniques required to construct a graphical user interface.  Can interact with screen or UI designers to ensure the logic layer integrates with the user interface  Can develop user interface coding and implementation - techniques for building user interfaces – for at least one channel  Can interact with testers to optimise the user interface |
| Data: can effectively link code to the database/data sets | Apprentices can link to a range of database types and embed data queries within their code  Can make a connection to a database Can execute CRUD statements on the database  Can use one-off queries and stored procedures  Can transform returned data in to format the application requires |
| Test: can test code and analyse results to correct errors found using either V-model manual testing and/or using unit testing | Apprentices can test and analyse their code to identify errors as soon as possible in the coding process and on an interactive basis  Can apply test and debugging strategies Can design and develop manual or unit tests Can test code segment functionality against requirements  Can assess test results against expected results and acceptance criteria |
| Problem solving  Problem solving: can apply structured techniques to problem solving, can debug code and can understand the structure of programmes in order to identify and resolve issues | Can use a minimum of two problem solving tools and techniques to identify and resolve programming issues  Can apply structured problem solving methods  Can apply problem-solving techniques to programming activities |
| Design: can create simple data models and software designs to effectively communicate understanding of the program, following best practices and standards | Can take a high level design and can interpret and convert the design in to simple data models and/or programme modules to communicate it to others  Can apply a software design methodologies (e.g., structured or object-oriented)  Can use standard design notation such as UML Can apply data modelling Can apply reconcile design against analysis models  Can design software solutions to meet requirements |
| Analysis: can understand and create basic analysis artefacts, such as user cases and/or user stories | Can take a variety of data and business requirements and convert them in to basic analysis artefacts to understand and can clarify the intended use of the proposed software  Can identify and represent required functionality (e.g. use cases)  Can identify and represent activity workflow (e.g. activity diagrams)  . |
| Deployment : can understand and utilise skills to build, manage and deploy code into enterprise environments | Can package and build completed programmes as appropriate to the resources available for deployment and for migration to different environments, including    developing appropriate user documentation  planning for user training and data migration. |
| Development lifecycle: can operate at all stages of the software development lifecycle, with increasing breadth and depth over time with initial focus on build and test. | Apprentices can operate as software developers showing a good understanding of the other phases of the software development lifecycle and the deliverables that are produced at each stage and as relevant to the development methodology (waterfall, agile, test led etc)  In addition to the stages above, the apprentice can also operate in the support and maintenance phases  Can advise third line support for relevant applications  Can fix bugs and deal with change requests |
| Can apply good practice approaches according to the relevant paradigm (for example object oriented, event driven or procedural) | Apprentices can identify and follow standards and good practice that can improve programming efficiency, style and quality, including  programming standards, both organisational and external  generic best practices including readability, reusability, maintainability  best practice approaches of different paradigms and languages |
| Can interpret and follow  software designs, functional/technical specifications,  company defined ‘coding standards’ or industry good practice for coding,  testing frameworks and methodologies, and  company, team or client approaches to continuous integration, version and source control | Apprentices can adapt to the employers domain and context for software development and interpret and follow the software development approach being implemented  Can read software designs and functional/technical specifications, especially those based upon the employer domain and context  Can identify, interpret and follow ‘coding standards’  Can identify, interpret and follow best practice coding approaches for specific paradigms and languages  Can identify, interpret and follow company, team or client approaches to continuous integration, version and source control |

|  |  |
| --- | --- |
| Can understand and respond to the business environment and business issues related to software development | Apprentices can apply the following considerations when working on projects  Business context  Business drivers (efficiency gains, increased functionality and improved quality of outputs) |
| Apprentices can understand and operate effectively in their own businesses, their customers, and the industries environments | Can demonstrate working within operational requirements such as health and safety, budgets, brands and normal business protocols |
| Apprentices understand the importance of service level agreements and/or agreed response times and operates within these |  |

## The how: the way in which the work has been done

|  |  |  |
| --- | --- | --- |
| **Competency Standard** | **Minimum expected requirements** | |
| Apprentices can demonstrate the full range of skills, knowledge and behaviours required to fulfil their job role  Apprentices can demonstrate how they contribute to the wider business objectives and show an understanding of the wider business environments | 1. The job:   Understanding the scope of the job role  Knowing what skills, knowledge and behaviours are needed to do the job well  Being aware of their own strengths in the job role, and any areas for improvement  Appreciating who else is important, for them to do their job and fulfil the role effectively (e.g. colleagues, managers, other stakeholders)  Being aware of potential risks in the job role (e.g. security, privacy, regulatory)  Using personal attributes effectively in the role, e.g. entrepreneurship  Understanding how the job fits into the organisation as a whole  Knowing what the next steps in their career might be   1. The organisation:   Understanding the goals, vision and values of the organisation  Knowing how they contribute to these in their own work  Being aware of the commercial objectives of the tasks/ projects they are working on  Understanding the importance of meeting or exceeding customers’ requirements and expectations  Being in tune with the organisation’s culture  Knowing how the organisation works, including its informal culture, internal networks etc.   1. The environment:   Being aware of the position of the organisation in the economy and its contribution to society  Understanding the key external factors that shape the way the organisation function, e.g. regulation  Knowing how the organisation can gain advantage in the industry, e.g. through innovation, technology, customer service etc. | |
| Apprentices can demonstrate the ability to use both logical and creative thinking skills when undertaking work tasks, recognising and applying techniques from both.  Apprentices can show that they recognise problems inherent in, or emerging during, work tasks, and can tackle them effectively. | a) Logical thinking:  Understanding initial premise(s) and preconditions  Analysing situations from known facts  Recognising the conclusion to be reached  Proceeding by rational steps  Evaluating information, judging its relevance and value  Supporting conclusions, using reasoned arguments and evidence  b) Creative thinking:  Looking at situations from a fresh perspective  Exploring ideas and possibilities  Making connections between different aspects  Questioning assumptions  Generating solutions that may be imaginative or unconventional  Devising new approaches  Adapting ideas and approaches as conditions or circumstances change  c) Problem-solving:  Analysing situations  Defining goals  Developing solutions  Prioritising actions  Dealing with unexpected occurrences |

## The with whom: the personal and interpersonal qualities the apprentice has brought to internal and external relationships

|  |  |
| --- | --- |
| **Competency Standard** | **Minimum expected requirements** |
| Apprentices can manage relationships with work colleagues, including those in more senior roles, customers/clients and other stakeholders, internal or external and as appropriate to their roles, so as to gain their confidence, keep them involved and maintain their support for the task/project in hand.  Apprentices can establish and maintain productive working relationships, and can use a range of different techniques for doing so. | 1. Managing relationships:   Understanding the value and importance of good relationships  Adopting a way of working that maintains and improves relationships  Involving other people in decisions and actions  Influencing others by listening to and incorporating their ideas and views  Acknowledging other people’s accomplishments and strengths  Overcoming barriers that prevent productive action  Managing conflict constructively  Promoting teamwork by encouraging others to participate   1. Customer/client relationships:   Establishing contact with customers/clients and keeping in touch  Keeping customers/clients informed (‘no surprises’)  Understanding their requirements, including constraints and limiting factors  Setting reasonable expectations  Involving them in decisions and actions (‘co-production’)  Interacting positively with them  Communicating in different ways  Providing a complete answer in response to queries (‘transparency’, ‘full disclosure’)  Actively seeking feedback   1. Stakeholders   Understanding who they are and what their ‘stake’ is  Prioritising stakeholders in terms of their importance, power to affect the task and interest in it  Using stakeholders’ views to shape projects early on  Gaining support from stakeholders, e.g. to win resources  Agreeing objectives  Managing expectations |
| Apprentices can communicate effectively with a range of people at work, one-to-one and in groups, in different situations and using a variety of methods.  Apprentices can demonstrate various methods of communication, with an understanding of the strengths, weaknesses and limitations of these, the factors that may disrupt it, and the importance of checking other people’s understanding. | a. Intention/purpose:  Understanding the purpose of communicating in a particular situation or circumstance (e.g. inform, instruct, suggest, discuss, negotiate etc.)  Checking that the person/people with whom one is communicating also understand the purpose  Being sensitive to the dynamics of the situation  Being aware of anything that might disrupt the effectiveness of the communication (e.g. status, past history)  b. Method:  Knowing the range of possible communication methods (e.g. spoken, written, graphical, multimedia)  Choosing a good, appropriate method for the situation  Being aware of the limitations of the chosen method, and the possible risks of miscommunication (e.g. ambiguity)  Taking account of the affective dimensions of the method (e.g. body language, tone of voice, eye contact, facial expression etc.)  c. Execution:  Expressing oneself clearly and succinctly, but not over- simplifying  Checking that the other person/people understand what is being expressed  Taking account of the potential barriers to understanding (e.g. filtering, selective perception, information overload)  Modifying the purpose and methods of communication during a situation in response to cues from the other person/people |