## Software Developer Template 5 – Declaration and Evidence Checklists for the Completion of the Summative Portfolio

##

These templates are provided to support the training provider in working with the apprentice and employer to ensure the successful completion of the summative portfolio.

The checklists can be used by training providers to help them manage the process through to completion, although training providers may also substitute their own processes and documentation as they see fit.

The key responsibilities of the apprentice in producing their summative portfolio can be found in the General Guidance for Apprentices, Employers and Training Providers, as can generic guidance on how to select evidence to compile the summative portfolio.

The apprentice should gather artefacts and record information that can evidence their activities undertaken in the workplace. The portfolio of evidence should demonstrate the full range of competencies, as shown in this template, which are required by the standard to show that the apprentice can fulfil the role of a software developer.

**Summative Portfolio Declaration**

**Apprentice declaration**

|  |  |
| --- | --- |
| Name |  |
| ULN |  |
| Declaration | I confirm that all the evidence submitted is my own work and it has been completed as specified. |
| Signature |  |
| Date |  |

**Line manager declaration (employer)**

|  |  |
| --- | --- |
| Name |  |
| Company |  |
| Declaration | I confirm that the work contained within this portfolio has, to the best of my knowledge, been completed solely by \_\_\_\_\_\_\_\_\_\_\_\_\_\_  |
| Signature |  |
| Date |  |

**Training provider declaration (training provider)**

|  |  |
| --- | --- |
| Name |  |
| Company |  |
| Declaration | I confirm that the work contained within this portfolio has, to the best of my knowledge, been completed solely by \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Signature |  |
| Date |  |

**Software developer technical competencies evidence checklist**

The defined competence areas and associated typical evidence are listed in this table. Not all employer businesses are identical so there will always be variation in the types of activity that will be carried out in the course of each apprentice’s daily work; however, each software development apprentice must be able to demonstrate evidence of every competence.

|  |
| --- |
| Competence |
| LogicWrite good quality code (logic) with sound syntax in at least one language. |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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.
 |  |  |

|  |
| --- |
| Competence |
| User interfaceCan develop effective user interfaces for at least one channel |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Evidence showing that you can apply the fundamental concepts of human–computer interaction or user experience design to program a high-quality 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 |  |  |

|  |
| --- |
| Competence |
| DataCan effectively link code to the database / data sets  |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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 |  |  |

|  |
| --- |
| Competence |
| TestCan test code and analyse results to correct errors found using either V-model manual testing and / or using unit testing |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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 |  |  |

|  |
| --- |
| Competence |
| Problem SolvingCan apply structured techniques to problem solving, can debug code and can understand the structure of programmes in order to identify and resolve issues |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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 |  |  |

|  |
| --- |
| Competence |
| DesignCreate simple data models and software designs to effectively communicate understanding of the program, following best practices and standards |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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 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 |  |  |

|  |
| --- |
| Competence |
| AnalysisCan understand and create basic analysis artefacts, such as use cases and / or user stories |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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) |  |  |

|  |
| --- |
| Competence |
| DeploymentCan understand and utilise skills to build, manage and deploy code into enterprise environments |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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
* data migration.
 |  |  |

|  |
| --- |
| Competence |
| Development life cycleCan operate at all stages of the software development life cycle, with increasing breadth and depth over time, with initial focus on build and test |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| 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) |  |  |
| 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 |  |  |

|  |
| --- |
| Competence |
| Interpret and FollowCan 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
 |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Apprentices can adapt to the employer’s 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 |  |  |

|  |
| --- |
| Competence |
| Respond to Business EnvironmentCan respond to the business environment and business issues related to software development |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Apprentices can apply the following considerations when working on projects:* Business context
* Business drivers (efficiency gains, increased functionality and improved quality of outputs)
 |  |  |

|  |
| --- |
| Competence |
| Industry EnvironmentApprentices can operate effectively in their own businesses, their customers, and the industries environments  |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Undertakes all work in accordance with agreed safety, technical and quality standards, using appropriate methods and tools. |  |  |

|  |
| --- |
| Competence |
| Application of mathematicsCan apply the maths required to be a software developer (e.g. algorithms, logic and data structures) |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Applies the maths required during Software Development Projects.  |  |  |

**Generic levels of responsibility evidence checklist**

Areas of responsibility and associated typical evidence are shown below.

|  |
| --- |
| Proficiency |
| Business skillsDemonstrates an analytical and systematic approach to issue resolution. Takes the initiative in identifying and negotiating appropriate personal development opportunities. Demonstrates effective communication skills. Contributes fully to the work of teams. Plans, schedules and monitors own work (and that of others where applicable) competently within limited deadlines and according to relevant legislation, standards and procedures. Appreciates the wider business context, and how their role relates to other roles and to the business of the employer or client. |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Evidence that you can demonstrate an analytical and systematic approach to issue resolution. |  |  |
| Evidence that you can take the initiative in identifying and negotiating appropriate personal development opportunities.  |  |  |
| Evidence that you can demonstrate effective communication skills. |  |  |
| Evidence that you can contribute fully to the work of teams.  |  |  |
| Evidence that you can plan, schedule and monitor own work (and that of others where applicable) competently within limited deadlines and according to relevant legislation, standards and procedures.  |  |  |
| Evidence that you can appreciate the wider business context, and how your role relates to other roles and to the business of the employer or client. |  |  |

|  |
| --- |
| Proficiency |
| Complexity Performs a range of work, sometimes complex and non-routine, in a variety of environments. Applies a methodical approach to issue definition and resolution. Undertakes all work in accordance with agreed safety, technical and quality standards, using appropriate methods and tools. |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Evidence that you can perform a range of work, sometimes complex and non-routine, in a variety of environments. |  |  |
| Evidence that you can apply a methodical approach to issue definition and resolution. |  |  |
| Evidence that you undertake all work in accordance with agreed safety, technical and quality standards, using appropriate methods and tools. |  |  |

|  |
| --- |
| Proficiency |
| AutonomyWorks under general direction. Uses discretion in identifying and responding to complex issues and assignments. Usually receives specific instructions and has work reviewed at frequent milestones. Determines when issues should be escalated to a higher level. |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Evidence that you can work under general direction. |  |  |
| Evidence that you can use discretion in identifying and responding to complex issues and assignments. |  |  |
| Evidence that you can usually receive specific instructions and have work reviewed at frequent milestones.  |  |  |
| Evidence that you can determine when issues should be escalated to a higher level. |  |  |

|  |
| --- |
| Proficiency |
| InfluenceInteracts with and influences colleagues. Has working level contact with customers, suppliers and partners. May supervise others or make decisions which impact the work assigned to individuals or phases of projects. Makes decisions which influence the success of projects and team objectives. |
| Minimum expected requirement | **List the evidence in the portfolio that fulfils this requirement** | **Reflections on applying knowledge learnt** |
| Evidence that you can interact with and influence colleagues. |  |  |
| Evidence that you have working level contact with customers, suppliers and partners. |  |  |
| Evidence that you may supervise others or make decisions which impact the work assigned to individuals or phases of projects. |  |  |
| Evidence that you can make decisions which influence the success of projects and team objectives. |  |  |