Accelerate People

Qualification Specification

Accelerate People L3 EPA for Software Development Technician

ST0128/AP03

QAN: 610/2026/6

Classification: Public

Version History		
Version	**Date Amended**	**Changes Made**
1.0	February 2023	Document created.
2.0	February 2025	Document rebranded. Edited gateway requirements to accommodate change in policy for English and Maths requirements.

Accelerate People

Contents

Qualification Objective

The level 3 Software Development Technician apprenticeship is one of a suite of digital apprenticeships that have been designed by industry employers to meet a range of job roles across different industries and sizes of business.

Accelerate People are an end-point assessment organisation (EPAO) for the digital apprenticeship standards that are defined by the Institute for Apprenticeships & Technical Education (IfATE). The apprenticeship standard and assessment plan can be found on the IfATE website.

As part of this apprenticeship all apprentices are required to complete an independent end-point assessment (EPA). The purpose of the EPA is to independently assess that any apprentice on this standard is competent in a relevant job role and can evidence meeting all the assessment criteria relating to the knowledge, skills and behaviours (KSB) outcomes.

The Level 3 Software Development Technician Apprenticeship

Role Profile:

This occupation is found in every sector in organisations ranging from large multi-nationals, public sector bodies and government projects developing multi-billion-pound software solutions to support key projects to small consultancy firms designing bespoke software solutions for clients.

For example, Financial Services, Transport, Security and Defence. The broad purpose of the occupation is to understand a client's requirements as provided in design specification and then build and test high-quality code solutions to deliver the best outcome.

Software development technicians are the supportive entry level team member helping to create computer programs. Some assist in developing the applications that allow people to do specific tasks on a computer or another device. Others assist in developing the underlying systems that run the devices or that control networks. For example, a software development technician may work to support a software developer or wider team on Transport ticketing systems, traffic light control systems, customer-facing

websites for journey planning and account management, internal websites for monitoring the status of train and road networks. They may assist in developing software to create bespoke asset management systems.

The software development technician may work on assisting software developer teams in devising innovative solutions to problems such as flood warning systems and creating products that enhance farmers engagement with sustainable farming approaches. Organisations use software to ensure that their operations become ever more effective and robust reducing the incidence of downtime by building quality tested software solutions to give a better service. For example, in commercial organisations this can give them a competitive advantage by being able to analyse significant amounts of data quickly and efficiently to provide the business with information and management systems. This can save time and help the business spot profit making opportunities. For public sector bodies the right software solution can drive up performance and help target scarce resources more effectively and ensure that customer expectations are more likely to be met.

A software development technician typically works as a junior member of a software development team, to build simple software components (whether web, mobile or desktop applications) to be used by other members of the team as part of larger software development projects or by end users. They will interpret simple design requirements for discrete components of the project under supervision. The approach will typically include implementing code, building on code that other team members have developed, to produce the required component. The software development technician will also be engaged in testing that the specific component meets its intended functionality. In their daily work, an employee in this occupation interacts with software developers and may also assist the wider team in their interactions with internal and external parties including users/customers (to understand their needs and evaluate the software developed through user testing). The software development technician may also interact under supervision with team members from a range of specialist fields including designers, developers, engineers, analysts, and project/delivery managers (to ensure the effective implementation of software solutions).

A software development technician is typically office-based however field-based research and testing may require periods of time working in the environments of the clients whose needs they are seeking to meet. An employee in this occupation will be responsible for assisting in the development of software solutions across the full software development life cycle, from research and development, through continuous improvement, to product/service retirement.

They will work under supervision on standalone project stages and as part of wider teams, reporting to a more senior member of their team.

Typical Job Titles:

Assistant programmer and automated test developer, junior application developer, junior application support analyst, junior developer, junior games developer, junior mobile app developer, junior programmer, junior software developer, junior web developer, software development technician.

Duties:

This apprenticeship standard includes duties to support alignment between the job role and the apprenticeship standard. Listed below are the duties that all apprentices must demonstrate in their apprenticeship. These duties are not assessed or graded as part of the EPA.

Duty 1: Follow clearly defined requirements to deliver software development activities and products.

Duty 2: Report progress against metrics on software development activities accurately throughout the stages of the software development lifecycle.

Duty 3: Identify and report any impediments to progress in development activities to supervisors.

Duty 4: Follow instructions to convert customer requirements to technical requirements.

Duty 5: Communicate outcomes from development activities to team members and other stakeholders both internal and external.

Duty 6: Identify and implement security features of a proposed design.

Duty 7: Write logical and maintainable software solutions in line with given specifications to meet the design requirements and organisational coding standards.

Duty 8: Apply security principles and practice to the software development tasks assigned.

Duty 9: Maintain appropriate project documentation throughout the software development tasks.

Duty 10: Apply appropriate recovery techniques to ensure that the software solution being developed is not lost.

Duty 11: Undertake unit testing of solutions, with appropriate levels of test code coverage, to identify and, where necessary, escalate issues.

Duty 12: Contribute to testing of the end-to-end software solution to ensure a high-quality output.

Duty 13: Support delivery of deployment phases, including trials and final release.

Duty 14: Identify the need for a suitable 'bug fix', appropriate to the severity and priority of the issue identified.

Duty 15: Practice continuous guided self-learning to keep up to date with technological developments to enhance relevant skills and take responsibility for own professional development.

Entry Requirements

Qualifications

Apprentices aged 16-18 on their apprenticeship start date, without level 2 English and maths, will need to achieve this level prior to taking the EPA. For those with an education, health and care plan or a legacy statement, the apprenticeship's English and maths minimum requirement is Entry Level 3. A British Sign Language (BSL) qualification is an alternative to the English qualification for those whose primary language is BSL.

Apprentices aged 19+ on their apprenticeship start date, without level 2 English and maths, are exempt from achieving this prior to taking their EPA; this exemption is by prior agreement between the apprentice and employer.

Experience

There are no pre-requisite knowledge, skills or understanding requirements defined for entry onto this qualification.

EPA Requirements

To successfully complete the level 3 Software Development Technician apprenticeship apprentices must achieve at least a pass in both EPA assessment methods. This EPA consists of two discrete assessment methods which have the following grades awarded.

Assessment Method 1: Project report with questioning.

- Pass.
- Distinction.
- Fail.

Assessment Method 2: Professional discussion underpinned by portfolio.

- Pass.
- Distinction.
- Fail.

All assessment methods must be taken within a three-month period, otherwise the entire EPA will need to be re-sat/re-taken.

EPA Gateway

For this apprenticeship all apprentices must spend a minimum of 12 months on programme, of which a minimum of 20% must be spent undertaking off-the-job training, before being eligible to undertake the EPA.

Before starting the EPA, an apprentice must meet the following gateway requirements:

- The employer is satisfied that the apprentice is working at or above the occupational standard.
- Apprentices must have achieved English and Maths qualifications in line with the apprenticeship funding rules.
- Apprentices must have compiled and submitted a portfolio of evidence to underpin the professional discussion.

If required, apprentices may request a reasonable adjustment. Information on how and when to apply is contained within the reasonable adjustments policy.

Once the apprentice is ready to enter gateway the following must be submitted to progress:

- Gateway form:
  - Confirming any dates the apprentice is unavailable during the EPA period.
  - Advising Accelerate People if the apprentice requires any reasonable adjustments to be made during the EPA.
  - Confirmation signatures that the apprentice is ready for the EPA.
- Evidence of:
  - Maths and English qualifications at Level 2 or above (or acceptable equivalent as specified in the entry requirements section), **or**
  - Confirmation that the apprentice is exempt from achieving English and Maths qualifications.
- The apprentices completed electronic portfolio (for AM2).

The gateway form along with all documentation must be uploaded before the EPA can commence. Failure to upload any of the required documentation may delay the EPA start date.

Knowledge, Skills and Behaviours

There are no mandatory vendor qualifications or knowledge modules for this apprenticeship. Apprentices are expected to be able to demonstrate competence against the assessment criteria specified within the assessment plan. The assessment criteria are based on the following KSBs, which apprentices are expected to be competent in before entering gateway.

Knowledge

K1: Fundamentals of all stages of the software development life cycle including development, Quality Assurance, User Acceptance Testing and release.

K2: Roles and responsibilities within the software development life cycle.

K3: Roles and responsibilities of the project life cycle.

K4: Different communication methods, how to adapt appropriately to different audiences and including collaborative technologies such as discussion threads and document collaboration.

K5: The key similarities and differences between different software development methodologies, such as agile and waterfall.

K6: Principles of effective teamwork to produce software.

K7: Fundamentals of software design approaches and patterns, including when to identify reusable solutions to commonly occurring problems.

K8: Organisational policies and procedures relating to the tasks being undertaken, and when to follow them. For example, the storage and treatment of General Data Protection Regulation (GDPR) sensitive data.

K9: Fundamentals of computing systems including physical, virtual and cloud technologies.

K10: Fundamental principles of algorithms, logic and data structures. For example, how they work using a step-by-step solution to a problem, or rules to follow to solve the problem.

K11: Principles and uses of relational and non-relational (NoSQL) databases.

K12: Basic principles of software designs and functional/technical specifications.

K13: Key principles of software testing frameworks and methodologies.

K14: Principles of pattern recognition such as looking for similarities among and within problems.

K15: Fundamentals of breaking down a complex problem or system into smaller, more manageable parts.

K16: The importance of valuing difference and understanding the protected characteristics named in the Equality Act 2010.

K17: Basic principles of emerging technology trends and innovations such as Internet of Things (IoT) Artificial Intelligence (AI) Augmented Reality (AR).

K18: Awareness of legal and regulatory requirements and their practical application to the role for example, Data Protection, Security, Intellectual Property Rights (IPR), Data sharing, marketing consent, personal data definition.

K19: Fundamental approaches to actions such as sequence, selection and iteration.

K20: The basic principles of software project planning including:

- Risks and dependencies.
- Integration.
- Prioritisation of tasks.
- Escalation of problems.
- Quality.
- Time.

- End user experience.

K21: Basic principles of processes and protocols used to ensure internet security, including concepts of security assurance.

K22: Key principles of testing for all components (including software, hardware, data), interfaces and the resulting service.

K23: Basic principles of Digital tools and their use in business:

- Management and presentation tools.
- Evaluation tools and techniques such as project management tools.

K24: Role and importance of Industry Standards and where to find them (e.g., ISO standards, IETF RFCs).

K25: Software development approaches for example object oriented, event driven or procedural.

Skills

S1: Write simple code for discrete software components following an appropriate logical approach to agreed standards (whether web, mobile or desktop applications) under supervision.

S2: Apply appropriate secure development principles to specific software components at all stages of development.

S3: Support development of effective user interfaces.

S4: Make simple connections between code and defined data sources as specified.

S5: Test simple code and analyse results to correct errors found using unit testing under supervision.

S6: Conduct a range of test types under supervision, such as Functional and Non-Functional.

S7: Apply structured techniques to problem solving, including carry out simple debug of code.

S8: Follows organisational and industry good coding practices (including for naming, commenting etc.).

S9: Solve logical problems, seeking assistance when required (including appropriate mathematical application).

S10: Support the creation of simple software documentation and visuals to effectively communicate understanding of the program.

S11: Define functional and non-functional requirements such as use cases, storyboards, user stories, performance and accessibility.

S12: Work within operational requirements such as health and safety, budgets, brands and normal business protocols.

S13: Develop user interfaces as appropriate to the organisations development standards and the type of software development being developed.

S14: Build scripts in line with work instructions for deployment into the relevant environment.

S15: Follow simple software designs and functional/technical specifications in line with work instructions.

S16: Follow simple testing frameworks and methodologies in line with work instructions.

S17: Follow company, team or client approaches to continuous integration, version and source control as instructed.

S18: Support the communication of software solutions and ideas to technical and non-technical stakeholders.

S19: Apply algorithms, logic and data structures in line with work instructions.

S20: Follow work instructions to contribute to building a given design whist remaining compliant with security and maintainability requirements.

S21: Apply techniques to break down complex problems.

S22: Demonstrate how Key Performance Indicators (KPIs) can be used to frame and measure desired outcomes.

S23: Implement secure code in appropriate languages of different types which is maintainable, readable, functional.

S24: Design simple software solutions to meet a requirement using tools and techniques, such as waterfall and agile.

S25: Work in a shared code base with appropriate etiquette and tools, such as modularity and data definition.

S26: Use simple debugging techniques, such as interactive debugging, print debugging, remote debugging.

S27: Implement test plans under supervision to show that a test plan is implementable in practice and implementation conforms to the plan.

S28: Develop and use simple acceptance criteria.

S29: Apply and maintain procedures and security controls to ensure confidentiality, integrity and availability.

S30: Use collaboration tools and technologies for source and version control to enable working together on common projects, regardless of physical location.

S31: Follow instructions to ensure client data is held securely under supervision e.g. not using personally identifiable information in test systems, making sure personal actions comply with ICO regulations.

S32: Use collaboration tools and technologies for writing technical documentation for, and adapting to, specific audience(s). e.g. technical, non-technical, internal, external.

Behaviours

B1: Use critical thinking skills when undertaking work tasks.

B2: Committed to guided Continuous Professional Development.

B3: Work independently and take responsibility within tightly defined parameters.

B4: Maintain a productive, professional and secure working environment.

B5: Team player, for example working collaboratively, keeping others informed using effective communication, recognising personal and professional limitations and seeking advice when necessary.

B6: Self-motivated, for example manages own time effectively, takes responsibility to complete the job.

Accelerate People

Assessment

AM1: Project Report with Questioning

The apprentice will conduct a project and deliver the outcome in the form of an electronic based report to the EPAO after a maximum of 4 weeks of the EPA start date. Apprentices will prepare their project report once they have passed the gateway. Following submission of the project, questioning will take place with an independent assessor.

Project Report

Whilst completing the project, the apprentice should be subject to normal workplace supervision.

The project may be based on any of the following:

- A specific problem.
- A recurring issue.
- An idea/opportunity.

The project report has a maximum word limit of 3,000, with a tolerance of plus or minus 10% (anything outside of this will be marked as a failure). Appendices, references, diagrams and/or video clips of up to 10 minutes in length are not included in this total (the video clip must be a file that can be uploaded, not a link to a video). The project must map (in an appendix) how it evidences the relevant KSBs for this assessment method as per the table below.

As a minimum all project reports must include:

- An introduction.
- The scope of the project (including key performance indicators).
- Analysis and problem solving in response to challenges within the project.
- Research and findings.
- Project outcomes including artefacts comprising examples of relevant coding undertaken and visual infographics conveying the software solution and design of the software development outputs sufficient to demonstrate the KSBs assigned to this method.
- Recommendations and conclusions.

- An explanation of how the stages of the software development lifecycle which are involved in the project have been evidenced e.g.
    - Planning.
    - Analysis.
    - Design.
    - Implementation/build.
    - Test.
    - Deploy.
    - Maintain.

Questioning

Questioning will involve questions that focus on the content of the project report. It is a structured conversation with an independent assessor and is designed to draw out the best of the apprentice's competence and excellence and covers the assessment criteria assigned to this assessment method.

Key points:

- Questioning will take place online via video conferencing.
- Apprentices will need access to the internet and a working webcam.
- The apprentice must have access to a quiet room and, unless reasonable adjustments have been requested for additional support, be alone in the room.
- Apprentices must have photographic identification (ID) to verify their identity, if they do not produce any ID then the questioning will be cancelled.
- The questioning will last for 40 minutes with the independent assessor having the discretion to increase the time of the questioning by up to 10% to allow the apprentice to complete their last answer.
- A minimum of 10 questions will be asked based on the project report and will be formed based on the evidence and grading requirements in the table below.
- Apprentices are allowed access to their project report throughout the questioning.
- Questions will only be based on the evidence submitted for this assessment method.
- The questioning cannot commence until a minimum of one week after the project report has been submitted.

AM2: Professional Discussion Underpinned by Portfolio

Portfolio

Training providers must work with the employer and apprentice to select the best evidence completed during the whole of the apprenticeship. All evidence should be real work tasks, and be clear, well documented and demonstrate competency against the assessment criteria listed in the assessment plan.

Typically, portfolios will contain ten discreet high-quality tasks covering a range of different assessment criteria in each, although it is expected that there will be overlaps of assessment criteria in each task.

Evidence sources may include:

- Written accounts of activities that have been completed.
- Photographic evidence and work products (annotated).
- Work instructions.
- Safety documentation.
- Technical reports.
- Drawings.
- Company policies and procedures as appropriate to the activities.
- Progress review documentation.
- Witness testimonies.
- Feedback from colleagues and/or clients.
- Video clips (maximum total duration 10 minutes); the apprentice must be always in view and identifiable.
- This is not a definitive list; other evidence sources are possible.

Where apprentices have worked on confidential or secure tasks, they should write high level statements about these tasks but not upload any restricted information or data. Apprentices should be prepared to discuss further details during the professional discussion.

Any employer contributions should focus on direct observation of performance (for example witness statements) rather than opinions. The evidence provided must be valid and attributable to the apprentice; the portfolio of evidence must contain a statement from the employer and apprentice confirming this.

The portfolio should **not** include any methods of self-assessment or standalone knowledge statements. Any demonstration of knowledge must be in context of a specific work-related task.

Portfolios should be in an electronic format which must be submitted to Accelerate People at gateway. Paper-based portfolios will not be accepted. If an apprentice uploads a video clip this must be a file that can be uploaded with their portfolio. A link to a video will not be accepted and will not be used as part of their evidence.

Professional Discussion

The professional discussion will take place at least two weeks after the portfolio has been accepted at gateway.

- The professional discussion will take place online via video conferencing.
- Apprentices will need access to the internet and a working webcam for the entire duration.
- The apprentice must have access to a quiet room and, unless reasonable adjustments have been requested for additional support, be alone in the room.
- Apprentices must have photographic identification (ID) to verify their identity, if they do not produce any ID then the professional discussion will be cancelled.
- The discussion will last for 60 minutes with the independent assessor having the discretion to increase the time of the questioning by up to 10% to allow the apprentice to complete their last answer.
- A minimum of 15 questions will be asked and will be formed based on the evidence and grading requirements in the table below.
- Apprentices are allowed access to their portfolio throughout the discussion.

Assessment Criteria

AM1: Project Report with Questioning

KSBs and Theme	Pass Criteria	Distinction Criteria
Software Development Lifecycle K1, K2, K3, K20, K23.	Explains the fundamentals of all stages of the software development life cycle in their project. (K1) Identifies roles and responsibilities within the software development	Evaluates the secure development principles used within the project by reference to specific software components throughout the stages of development. (S2)

KSBs and Theme	Pass Criteria	Distinction Criteria
S2, S15, S22. B3.	life cycle and contrasts them with the roles and responsibilities of the project life cycle. (K2, K3) Demonstrates the basic principles of software project planning including: Risks and dependencies, integration, prioritisation of tasks, escalation of problems, quality, time and end user experience. (K20) Demonstrates the basic principles of digital tools and their use in business such as: • Management and presentation tools. • Evaluation tools and techniques such as project management tools. (K23) Applies appropriate secure development principles to specific software components at all stages of development in the project. (S2) Demonstrates following simple software designs and functional/technical specifications in line with work instructions. (S15) Demonstrates how Key Performance Indicators (KPIs) can be used to frame and measure desired outcomes. (S22) Demonstrates how they work independently and to take responsibility within tightly defined parameters. (B3)	Evaluates the principles followed, and the tools used in the project planning and development stages and how the tools support organisational performance. (K20, K23)
Software Testing	Explains the key principles of software testing frameworks and methodologies. (K13)	Critically analyses testing frameworks and methodologies. (K13, S7)

KSBs and Theme	Pass Criteria	Distinction Criteria
K13. S5, S7, S16, S26.	Explains how they test simple code and analyse results to correct errors found using unit testing under supervision. (S5) Explains how they apply structured techniques to problem solving, including carry out simple debug of code. (S7) Explains how they have followed simple testing frameworks and methodologies in line with work instructions. (S16) Describes simple debugging techniques, such as interactive debugging, print debugging, remote debugging. (S26)	
Development K7, K10, K12, K14, K15, K19. S1, S11, S14, S19.	Explains the fundamentals of software design approaches and patterns, including when to identify reusable solutions to commonly occurring problems in the project. (K7) Describes the fundamental principles of algorithms, logic and data structures. (K10) Identifies basic principles of software designs and functional/technical specifications within the project. (K12) Describes the principles of pattern recognition such as looking for similarities among and within problems. (K14) Demonstrates the fundamentals of breaking down a complex	

KSBs and Theme	Pass Criteria	Distinction Criteria
	problem or system into smaller, more manageable parts. (K15) Utilises fundamental approaches to actions such as sequence, selection and iteration. (K19) Writes simple code for discrete software components following an appropriate logical approach to agreed standards under supervision. (S1) Explains how they have defined functional and non-functional requirements such as use cases, storyboards, user stories, performance and accessibility. (S11) Explains building scripts in line with work instructions for deployment into the relevant environment. (S14) Applies algorithms, logic and data structures in line with work instructions. (S19)	
Software Support S10. B4, B6.	Describes how they support the creation of simple software documentation and visuals to effectively communicate understanding of the program. (S10) Explains how they maintain a productive, professional and secure working environment. (B4) Explains how they are self-motivated, for example manages own time effectively, takes responsibility to complete the job. (B6)	

AM2: Professional Discussion Underpinned by Portfolio

KSBs and Theme	Pass Criteria	Distinction Criteria
Systems and Support K9, K11, K17. B2.	Explains the fundamentals of computing systems including physical, virtual and cloud technologies. (K9) Explains the principles and uses of relational and non-relational (nosql) databases. (K11) Outlines the basic principles of emerging technology trends and innovations. (K17) Reflects on their progress in the organisation with reference to their CPD record and identifies areas for future development/participation. (B2)	Analyses emerging technology trends and innovations. (K17)
Software Testing K22. S6.	Describes the principles of testing for components, interfaces, and the resulting service. (K22) Describes how they have conducted a range of test types under supervision. (S6).	
Working Legally and Securely K8, K16, K18, K21, K24. S8, S12, S20, S23, S29, S31.	Describes the organisational policies and procedures relating to the tasks being undertaken, and when to follow them. (K8) Outlines the importance of valuing difference and understanding the protected	Evaluates their contribution to ensuring software design solutions include secure principles. (K21, S20, S23, S29) Critically evaluates the importance of coding standards (organisational or industry) in a team

KSBs and Theme	Pass Criteria	Distinction Criteria
	characteristics named in the Equality Act 2010. (K16) Explains legal and regulatory requirements and outlines their practical application to the role of software developer. (K18) Explains the basic principles of processes and protocols used to ensure internet security. (K21) Explains the role and importance of Industry Standards and where to find them (e.g. ISO standards, IETF RFCs). (K24) Explains how they follow organisational and industry good coding practices (including for naming, commenting etc.) (S8) Explains how they work within operational requirements such as health and safety, budgets, brands and normal business protocols. (S12) Justifies their interpretation and implementation of work instructions to contribute to building a given design whist remaining compliant with security and maintainability requirements. (S20) Explains how they have implemented secure code in languages of different types which is maintainable, readable and functional. (S23)	environment as well as working individually with reference to their projects. (S8)

KSBs and Theme	Pass Criteria	Distinction Criteria
	Outlines how they apply and maintain procedures and security controls to ensure confidentiality, integrity and availability. (S29) Explains how they follow instructions to ensure client data is held securely under supervision. (S31)	
Development K5, K25. S3, S4, S9, S13, S17, S21, S24, S25, S27, S28, S30, S32. B1.	Outlines the key similarities and differences between different software development methodologies. (K5) Compares and contrasts different software development approaches for example object oriented, event driven or procedural. (K25) Describes how they have supported the development of effective user interfaces. (S3) Makes simple connections between code and defined data sources as specified. (S4) Explains how they solve logical problems, seeking assistance when required. (S9) Demonstrate how they have developed user interfaces as appropriate to the organisation's development standards and the type of software development being developed. (S13) Explains how they follow company, team or client approaches to continuous	Evaluates the different software development methodologies and justifies the choice of methodology used with reference to their projects and the organisation. (K5) Analyses different approaches to user interface development with reference to their organisations functional/technical standards and software type, justifying their choice for the project'. (S13) Analyses techniques to break down complex problems identifying how approaches to their application vary dependent on the software context. (S22) Evaluates their contribution to building a given software design solution and their approach to working collaboratively. (K6, S17, S25, S30)

KSBs and Theme	Pass Criteria	Distinction Criteria
	integration, version and source control as instructed. (S17) Discusses how they apply techniques to break down complex problems. (S21) Explains how they design simple software solutions to meet a requirement justifying their choice of tools and techniques. (S24) Describes how they work in a shared code base using selected tools and following the defined etiquette and tools. (S25) Explains how they implement test plans under supervision to show that it is implementable in practice and conforms to the plan. (S27) Explains how they develop and use simple acceptance criteria. (S28) Outlines how they use collaboration tools and technologies for source and version control to enable working together on common projects, regardless of physical location and for writing technical documentation for, and adapting to, specific audience(s). (S30, S32) Reflects on their use of critical thinking skills when undertaking work tasks. (B1)	

KSBs and Theme	Pass Criteria	Distinction Criteria
Communication and Collaboration K4, K6. S18. B5.	Describes different communication methods, how to adapt in response to different audiences including collaborative technologies such as discussion threads and document collaboration. (K4) Outlines the principles of effective teamwork to produce software. (K6) Describes how they support the communication of software solutions and ideas to technical and non-technical stakeholders. (S18) Describes how they are a team player, for example working collaboratively, keeping others informed using effective communication, recognising personal and professional limitations and seeking advice when necessary. (B5)	

Grading

Each assessment method is graded individually and combined to give an overall grade. Assessment criteria do not appear in more than one assessment method, therefore an assessment criteria failed in one assessment method cannot then be demonstrated in the other assessment method. All EPA methods must be passed for the EPA to be passed overall.

Grades from individual assessment methods should be combined in the following way to determine the grade of the EPA as a whole:

Project Report with Questioning	Professional Discussion Underpinned by Portfolio	Overall Grading
Fail	Any grade	**Fail**
Any grade	Fail	**Fail**
Pass	Pass	**Pass**
Pass	Distinction	**Merit**
Distinction	Pass	**Merit**
Distinction	Distinction	**Distinction**

Re-sits and Re-takes

Apprentices who fail one or more assessment method will be offered the opportunity to take a re-sit or a re-take at the employer's discretion. The apprentice's employer will need to agree that either a re-sit or re-take is an appropriate course of action.

A re-sit does not require further learning, whereas a re-take does. Apprentices should have a supportive action plan to prepare for a re-sit or a re-take.

If the apprentice fails the project assessment method, they will be required to amend the project in line with the independent assessor's feedback. The apprentice will be given 3 weeks to rework and submit the amended project. The independent assessor will have 2 weeks to review the project.

The timescales for a re-sit/re-take is agreed between the employer and EPAO. A re-sit is typically taken within two months of the EPA outcome notification. The timescale for a re-take is dependent on how much re-training is required and is typically taken within three months of the EPA outcome notification.

All assessment methods must be taken within a three-month period, otherwise the entire EPA will need to be re-sat/re-taken.

Re-sits and re-takes are not offered to apprentices wishing to move from pass to a higher grade.

Where any assessment method must be re-sat or re-taken, the apprentice will be awarded a maximum EPA grade of distinction.

Specimen

All specimen materials can be accessed by registered training providers from the knowledge area on ACE360.

Accelerate People

Accelerate People are an independent EPAO. If you have any questions or queries relating to this qualification specification or EPA, please contact us using the details below.

Registered training providers with Accelerate People can access further guidance material on the knowledge base on ACE360.

Contact Details

Email: info@accelerate-people.co.uk

Visit: www.accelerate-people.co.uk

Registered office: Accelerate People Limited, Scale Space, Imperial College White City Campus, 58 Wood Lane, London, W12 7RZ.

Registered in England with number: 09577006.

Registered Apprenticeship Assessment Organisation Number: EPO 0475.


Accelerate People