Standard
L3: Software Development Technician

UOS reference number

ST0128_V1.1

Trailblazer reference number

TB0546

Title of occupation

Software Development Technician

Trailblazer name

Software Tester

Core and options

No

Resubmission

No

Level of occupation

Level 3

Route

Digital

Typical duration of apprenticeship

18 months

Target date for approval

No target date

Occupational profile

Summary

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 test 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

Typical job titles include Software Development Technician, Junior Developer, Junior Web Developer, Junior Application Developer, Junior Mobile App Developer, Junior Games Developer, Junior Software Developer, Junior Application Support Analyst, Junior Programmer, Assistant Programmer and Automated Test Developer.

Duties

Duty	Knowledge	Skills	Behaviours
D1: Follow clearly defined requirements to deliver software development activities and products	K1, K2, K3, K12, K13, K14, K20, K21, K23, K24	S2, S10, S12, S15, S16, S18, S25, S30, S33	B1, B2, B3, B6
D2: Report progress against metrics on software development activities accurately throughout the stages of the software development lifecycle	K1, K2, K3, K4, K5, K6, K20	S12, S15, S18, S19, S21, S26, S31, S33	B1
D3: Identify and report any impediments to progress in development activities to supervisors	K1, K2, K3, K4, K13, K15, K20, K23, K24	S1, S5, S6, S7, S9, S17, S18, S19, S22, S23, S25, S26, S33	B1, B3, B4
D4: Follow instructions to convert customer requirements to technical requirements	K1, K2, K4, K6, K23	S2, S3, S19, S32, S33	B4
D5: Communicate outcomes from development activities to team members and other stakeholders	K4, K18, K20, K23	S12, S19, S32, S33	B1, B4
D6: Write logical and maintainable software solutions in line with given specifications to meet the design requirements and organisational coding standards.	K6, K7, K9, K10, K11, K12, K13, K14,	S1, S2, S5, S14, S15, S16, S17,	
D7 Take the non-functional requirements of maintenance, performance and user experience into account along with the functional specification provided	K12	S6, S11	
D8: Apply security principles and practice to the software development tasks assigned, implement security best practices to ensure software is not vulnerable to malicious attacks	K1, K7, K13, K14, K15	S2, S5, S21, S24, S25	B2
D9: Maintain appropriate project documentation throughout the software development tasks	K1, K2, K3, K4, K20	S12, S18, S26, S31, S32, S33	—
D10 Apply appropriate recovery techniques to ensure that the software solution being developed is not lost. For example, work with source control tools to provide a record of changes to source code, share code with the team, and ensure code is safely stored for recovery	—	S26, S31, S33	B2, B4
D11: Undertake unit and integration testing of solution to meet code coverage guideline, reduce the number of defects, and provide confidence in the quality of the software	K1, K2, K13, K22	S5, S6, S17, S28	—
D12: Contribute to testing of the end-to-end software solution to ensure a high-quality output and where necessary escalate issues.	K1, K2, K13, K22	S5, S6, S17	B2, B3
D13: Provide support throughout the development lifecycle, including user acceptance testing and final release to production	K1, K2, K3, K5, K6, K9, K20, K24	S18, S19, S23, S30, S31, S32, S33	B4

Duties

D14: Provide initial support to classify severity and priority of issues and schedule bug fixes where necessary.	K1, K2, K6, K7, K10, K11, K12, K14, K15, K21, K22	S15, S16, S17, S18, S19, S20, S21, S22, S24, S27	B2, B3
D15: Practice continuous guided self-learning to keep up to date with technological developments to enhance relevant skills and take responsibility for own professional development	K16, K17, K18, K24	—	B1, B5

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 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: 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 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 such as 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 a given software development approach according to the relevant paradigm (for example object oriented, event driven or procedural) in line with work instructions DELETE

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 a supported context 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.

Duty	Training requirement	Method of delivery	Provider type	OTJ days
D1: Follow clearly defined requirements to deliver software development activities and products				20
D2: Report progress on software development activities accurately throughout the stages of the software development lifecycle				5
D3: Identify and report any impediments to progress in development activities to supervisors				2
D4: Follow instructions to convert customer requirements to technical requirements				5
D5: Communicate outcomes from development activities to team members and other stakeholders both internal and external				1
D6: Identify and implement security features of a proposed design				10
D7: Write logical and maintainable software solutions in line with given specifications to meet the design requirements and organisational coding standards.				10
D8: Apply security principles and practice to the software development tasks assigned				5
D9: Maintain appropriate project documentation throughout the software development tasks				5
D10: Apply appropriate recovery techniques to ensure that the software solution being developed is not lost.				10
D11: Undertake unit testing of solutions, with appropriate levels of test code coverage, to identify and, where necessary, escalate issues.				5
D12: Contribute to testing of the software solution to ensure a high quality output				5
D13: Support delivery of deployment phases, including trials and final release.				2
D14: Identify the need for a suitable 'bug fix', appropriate to the severity and priority of the issue identified.				5

Duty	Training requirement	Method of delivery	Provider type	OTJ days
D15: Practice continuous guided self-learning to keep up to date with technological developments to enhance relevant skills and take responsibility for own professional development				1

Additional information

Entry requirements

No entry requirements specified

Professional recognition

No professional body recognition specified

Trailblazer membership details

Chair

Rebecca Plant (Microsoft)

Facilitator

Max Reynolds (None)

Employer members

Name	Employer
Lisa Blows	IBM
Penny Wilsher	First Finance
Phil Vetter	Exclaimer
Mark Harrop	DEFRA
John Lockwood	FUJITSU

Other members

Name	Employer
Yonas Meressi	Firebrand Training