THE REGISTERED APPRENTICESHIP OCCUPATIONS AND STANDARDS CENTER OF EXCELLENCE (AOSC)

Software Quality Assurance Tester National Occupational Framework

ONET Code: 15-1253.00

RAPIDS Code: 3070

Created: February 2024

This project has been funded, either wholly or in part, with federal funds from the Department of Labor, Employment and Training Administration under Cooperative Grant Number AP-36653-21-75-A-11. The contents of this publication do not necessarily reflect the views or policies of the Department of Labor, nor does mention of trade names, commercial products, or organizations imply endorsement of the same by the US Government.

URBAN INSTITUTE · ELEVATE · THE · DEBATE

ABOUT THE URBAN INSTITUTE

The nonprofit Urban Institute is dedicated to elevating the debate on social and economic policy. For nearly five decades, Urban scholars have conducted research and offered evidence-based solutions that improve lives and strengthen communities across a rapidly urbanizing world. Their objective research helps expand opportunities for all, reduce hardship among the most vulnerable, and strengthen the effectiveness of the public sector.

Acknowledgments

We would like to thank several people who have contributed to the development and vetting of this National Occupational Framework. We would especially like to thank Diane Jones for her research, support, and contributions to the development of the framework. Additionally, we had terrific experts lend their time, review, and support to the framework. They include Susan Wright of Cloud for Good; Clifford Chipman and Ashley Parsons-Crowder of Franklin Apprenticeships; Andrew Pollner of American Software Testing Qualifications Board; Nicolas Patschkowski of Open Classrooms; Randall Rice of Association for Testing & Software Quality Assurance; and Terry Farrar of International Business Machines Corporation (IBM). We would like to extend a special thanks to Chris Kenst of Association for Software Testing and Gwen Iarussi of Xpanxion for their multiple rounds of review. We also want to thank Myca San Miguel, Shruti Nayak, and Bhavani Arabandi for their thoughtful creation and review of this framework. Finally, we want to thank our editors, Lexi Mills and Liza Hagerman.

Introduction to Using This Document

Under the Registered Apprenticeship Technical Assistance Centers of Excellence award, the Urban Institute leads the Occupations and Standards work. One of the main objectives of Urban's project is to create high-quality, well-researched, consensus-based work process schedules that are nonproprietary and widely available. This document is a product of that work and contains three sections: the occupational overview, the work process schedule, and the related technical instruction.

The **occupational overview** is a general introduction, including alternative job titles, any prerequisites, and, if applicable, the total number of hours needed to complete a time-based or hybrid program.

The **work process schedule** outlines the major job functions, competencies, and/or hours an apprentice completes in a registered apprenticeship program. It outlines what apprentices are expected to learn on the job with the support of a mentor or journeyworker (a worker mastering the competencies of an occupation in a particular industry), including both core competencies and those deemed optional by experts in the field. The work process schedule is the foundational document guiding a program.

Urban works with numerous experts to ensure the content is thoroughly researched and vetted to reflect the expectations of industry, educators, labor unions, employers, and others involved in apprenticeship for this occupation. Sponsors and employers can use the work process schedule as their program standards with assurances it has been approved by experts in the field.

The **related technical instruction** presents considerations for the coursework that apprentices will undertake to supplement on-the-job learning. It is intended to serve as a reference to sponsors exploring their options for the accompanying classroom, virtual, or hybrid training.

How to Use the Work Process Schedule

Sponsors can adapt the work process schedule to accommodate their needs for competency- or time-based or hybrid programs. In a **competency-based** apprenticeship, sponsors assess apprentices' progress across core and optional competencies listed in the work process schedule. In a **time-based** apprenticeship, apprentices complete a predetermined number of hours across major job functions and the program overall. In a **hybrid** apprenticeship, sponsors monitor apprentices' hours spent on major job functions and assess their proficiency across competencies.

Each program type has a different method of assessment:

- **For a competency-based program**, apprentices engage in activities and make progress toward proficiency in the identified competencies. Sponsors overseeing apprentices' work assess their mastery of the outlined competencies using the following rating scale:

  4—Competent/proficient (able to perform all elements of the task successfully and independently)

  3—Satisfactory performance (able to perform elements of the task with minimal assistance)

  2—Completed the task with significant assistance

  1—Unsuccessfully attempted the task

  0—No exposure (note the reason—absence, skill isn't covered, etc.)

  The competencies may be completed in any order. Apprentices must perform at a level 4 or 3 in all competencies listed as "core" to complete the apprenticeship program successfully.

- **For a time-based program,** sponsors monitor apprentices' completion of hours in training across major job functions. The total number of hours recommended for this occupation is listed in the occupational overview and is based on guidance from the US Department of Labor. Generally, apprentices must have at least 2,000 hours overall for on-the-job learning, but occupations of greater complexity may require more hours. Sponsors will provide apprentices with supervised work experience and allocate the total number of hours across the major job functions to adequately train their apprentices.

- **The hybrid approach** blends both competency- and time-based strategies. Sponsors measure apprentices' skills acquisition through a combination of completing the minimum number of hours of on-the-job learning successfully demonstrating identified competencies. Sponsors will assess apprentices' proficiencies as described for competency-based programs with a rating scale of 0–4 for every core competency. Generally, apprentices have at least 2,000 hours overall for on-the-job learning, but occupations of greater complexity may require more hours. Sponsors will document apprentices' completion within a minimum and maximum range of hours assigned for each major job function.

Software QA Tester Occupational Overview

Occupational Purpose and Context

Software quality assurance testers (QA Testers) ensure that the functionality, design, and usability of an application or website are operating in accordance with client or stakeholder needs. QA Testers support quality delivery outcomes by conducting manual and automated testing of an application throughout the software delivery lifecycle.

Potential Job Titles

Quality assurance engineer, quality engineer, test engineer, software quality assurance analyst, software quality assurance tester, quality assurance analyst, quality assurance tester for software, software tester, junior quality assurance engineer, test automation engineer, quality assurance automation engineer, software development engineer in test (SDET)

Apprenticeship Prerequisites

Apprentices may benefit from having a background in information technology and a basic understanding of computing technology and application systems, from a user interface perspective to a data level. They may also benefit from some knowledge of cloud environments, operating systems (Windows, Mac, Unix/Linux-based) and application platforms (web vs. native/mobile). Apprentices will benefit from being "systems thinkers," as they must connect how a particular functionality will impact or delay other processes of the product. They should be organized and possess good communication skills to provide feedback throughout the testing lifecycle. They should have a high attention to detail and be excellent at time management and prioritizing tasks.

Recommended Length of Apprenticeship (Time/Hybrid Programs Only)

Previous Background or Experience	Recommended Length of Apprenticeship
No background or experience	3,000–4,000 hours
Some previous background or experience	2,000–3,000 hours

Work Process Schedule

Instructions for Use:

Competency-based programs: In the "performance level achieved" column of the work process schedule (see examples starting on the next page), assess apprentices' performances on each competency with the scale below. No monitoring of hours is required for this approach. See "Guidelines for Competency-Based, Hybrid and Time-Based Apprenticeship Training Approaches," US Department of Labor, Employment and Training Administration, Office of Apprenticeship, October 20, 2015, https://www.apprenticeship.gov/sites/default/files/bulletins/Cir2016-01.pdf.

Software QA Tester

ONET Code: 15-1253

RAPIDS Code: 3070

4—Competent/proficient (able to perform all elements of the task successfully and independently)

3—Satisfactory performance (able to perform elements of the task with minimal assistance)

2—Completed the task with significant assistance

1—Unsuccessfully attempted the task

0—No exposure (note the reason—absence, skill isn't covered, etc.)

Time-based programs: In the "hours" row, specify the number of hours apprentices will fulfill for each job function. No assessment of competencies is required for this approach.

Hybrid programs: In the "performance level achieved" column, assess apprentices' performances on each competency using the 0–4 scale above. In the "hours" row, identify a range of hours apprentices should spend working on each major job function.

Job Function 1: Gathers and analyzes requirements to understand the role of testing

Hours (time-based and hybrid programs only):

Competencies	Core or optional	Performance level achieved (0–4) (competency-based and hybrid programs only)
A. Determines the objectives of testing under various conditions	Core	
B. Gathers and analyzes requirements in partnership with software developers and project team to translate them into testable criteria	Core	
C. Assesses functional requirements for impact, risk, and feasibility; flags areas for improvements where appropriate	Core	
D. Interprets nonfunctional requirements to meet business requirements for end level users	Optional	
E. Understands the relationship between testing and quality assurance and gives examples of how testing contributes to higher quality of software	Optional	

Job Function 2: Identifies test design and strategizes test plans

Hours (time-based and hybrid programs only):

Competencies	Core or optional	Performance level achieved (0–4) (competency-based and hybrid programs only)
A. Analyzes test deliverables, objectives, and past defects to inform current decision-making	Core	
B. Performs respective tasks within the test process	Core	
C. Follows test plans and test schedules	Core	
D. Utilizes existing test environments	Core	
E. Assesses the state of quality at any given time and supports advocating for quality improvements	Core	
F. Differentiates between whitebox and blackbox testing	Core	
G. Varies testing strategy based on mission and risk	Core	
H. Examines relationships between software development activities and test activities in the software development lifecycle	Core	

I. Applies static and dynamic techniques, considering objectives, types of defects to be identified, and the role of these techniques within the software lifecycle	Optional	
J. Recognizes types of software work product that can be examined by the different static testing techniques	Optional	
K. Assesses the benefits and risks of test automation	Optional	

Job Function 3: Collaborates with internal teams, clients, and stakeholders		
Hours (time-based and hybrid programs only):		
Competencies	**Core or optional**	**Performance level achieved (0–4)** (competency-based and hybrid programs only)
A. Fosters open communication through active listening and facilitates shared understanding across teams and clients	Core	
B. Offers and accepts feedback respectfully, meets others with empathy, and seeks additional feedback and perspectives	Core	
C. Participates in self-planning activities, prioritizes tasks applying Agile and/or Scrum principles (or similar), maintains knowledge of application life cycle management tools such as Atlassian stack and/or Jira (or similar)	Core	
D. Addresses challenges with curiosity, seeks to understand the root cause, and initiates follow-up conversations to provide insight and resolutions	Core	
E. Collaborates with engineering team on complex issues requiring escalation and effectively coordinates with teams	Core	
F. Abides by principles of inclusion and equity in communication with team members and superiors	Core	
G. Communicates data, findings, and solutions in a clear and concise manner to nontechnical individuals within the business	Optional	

Job Function 4: Conducts relevant testing (the competencies here assume functional and non-functional testing is applicable)		
Hours **(time-based and hybrid programs only):**		
Competencies	Core or optional	Performance level achieved (0–4) (competency-based and hybrid programs only)
A. Performs testing agreed upon in plan and iterates that testing based on results	Core	
B. Discerns between different types of testing (functional, regression, accessibility, usability, smoke system, etc.) and uses the appropriate test	Core	
C. Conducts exploratory testing when applicable	Core	
D. Conducts regression testing	Core	
E. Conducts exploratory testing	Core	
F. Assesses the state of quality at any given time and advocates for quality improvements	Core	
G. Considers how product risk analysis may influence the thoroughness and scope of testing	Core	
H. Varies tests design based on past successes, failures, and risk	Core	
I. Reviews traceability matrix to ensure tests are mapped to the requirements	Core	
J. Applies a review technique to a work product to find defects	Optional	
K. Conducts performance testing	Optional	
L. Assists in the selection and implementation process of testing tools	Optional	
M. Works with change management team to integrate QA tests into release pipelines, using tools such as Jenkins, ansible, etc. (only applicable for automation)	Optional	
N. Develops automated tests and ensures they are working properly	Optional	

Job Function 5: Monitors software functionality and reports on defects; supports post-release testing and activities		
Hours (time-based and hybrid programs only):		
Competencies	**Core or optional**	**Performance level achieved (0–4)** (competency-based and hybrid programs only)
A. Creates defect and incident reports based on company protocols	Core	
B. Participates in retrospectives (Agile) or post-release reviews as part of continuous improvement initiatives to improve testing over time	Core	
C. Collaborates with support and development in troubleshooting and resolving identified production issues	Core	
D. Adheres to company policies and methodologies for troubleshooting issues, identifying and replicating defects, and documenting and reporting changes	Core	
E. Writes and communicates clear and understandable defect reports	Core	
F. Remains up to date on emerging testing technologies and methodologies	Optional	
G. Researches and identifies remediations, collaborates with business analysts and developers to find solutions	Optional	
H. Monitors production release and responds to observability/application monitoring feedback	Optional	

Related Technical Instruction

Instructions for Use:

Registered apprenticeships must include at least 144 hours of related technical instruction (RTI). Courses offered by accredited colleges and universities may be assigned a credit hour determination rather than a contact hour determination. In general, an academic credit unit is the equivalent of 15 clock hours of instruction.

Development and Use of This RTI Outline: Employers and academic institutions may approach RTI in markedly different ways. Our goal was not to identify the single best way to provide RTI or to identify a single provider whose content we deemed to be superior. Instead, our goal was to survey numerous education providers, including employers, institutions of higher education, high schools, private continuing education providers, labor organizations, professional associations and, in some cases, municipalities that provide worker training, to identify topics or courses common among those providers that align with the job functions included in this work process schedule. Those common topics or courses are reflected in the RTI outline provided below, which may be useful in developing your RTI program or communicating your needs to an educational partner.

Licensure or certification requirements: N/A
Degree requirements for licensure or certification, if applicable: N/A
Accreditation requirements of instructional provider for licensure or certification, if applicable: N/A
Anticipated changes in licensure or certification requirements, if known: ISTQB certifications changed to allow remote electronic exams due to COVID, and those changes were made permanent.
Examples of state licensure or certification requirements: N/A

Examples of RTI providers for this occupation

Professional associations and labor organizations: The Association for Software Testing provides a variety of training opportunities, as does the International Institute for Software Testing. The International Institute for Software Testing also provides opportunities for testers to become Certified Software Test Professionals. The American Software Testing Qualifications Board offers the ISTQB Exam for Foundation, Agile, Advanced, & Specialist Software Testing Certifications. ISTQB also offers freely accessible resources, including a syllabus covering testing materials and glossary.

Military: Training is available to service members who work in computer science and computer engineering fields.

States/municipalities: None.

Colleges and universities: Several two-year and four-year colleges offer certificate programs in software testing. Bachelor's degree programs in computer science, software development and software engineering may include course offerings in software testing.

No-cost online providers: EdX and Coursera offer numerous courses in software testing. Many of these courses are taught by accredited colleges and universities in the US and abroad, and some are taught by private companies including Google, Amazon Web Services, and IBM.

Continuing education or specialty education providers: Numerous continuing education providers offer certificate programs, individual course options and bootcamps designed to prepare individuals to enter the field of software testing or to qualify for career advancement opportunities in the field.

Prerequisite knowledge, skills or experience typically required by RTI providers for this occupation

Apprenticeship sponsors and employers of software QA testers look for individuals who have some level of computer experience, strong skills in logic, and are detail-oriented.

Business Communication

Hours: 20–30

Sample learning objectives

- Demonstrate the ability to send and receive phone calls, emails, text messages, instant messages, and other forms of electronic communication.
- Demonstrate the ability to compose emails, formal letters, memoranda and reports using appropriate format, spelling, capitalization, grammar, and punctuation.
- Demonstrate the ability to provide detailed instructions verbally and in writing to explain how a particular process is done, how a product is made or to explain decision logic.

- Describe effective strategies for engaging in active listening and assessing whether your message is understood by another party.
- Demonstrate the ability to work as part of a team to create a report or complete a project.
- Demonstrate effective strategies for managing conflict and maintaining calmness and composure under stressful conditions.

Introduction to the Software Development Life Cycle

Hours: 20–30

Sample learning objectives

- Define the key steps in the software development life cycle and explain the role of software testers in each of these steps.
- Describe the most commonly used models for software development, such as Waterfall, Agile, Iterative, Rapid Application Development, Spiral, and Big Bang, and list the advantages and disadvantages of each.
- Evaluate client task orders to determine which software development model will best meet specific client needs.
- Explain what ISO/IED 12207 standards are and describe their role in improving software design and reliability.
- Differentiate between high-level and low-level design.
- Identify potential risks associated with each stage of the software development lifecycle that could reduce software capability, development costs or development timelines.

Introduction to Software Development

Hours: 20–30

Sample learning objectives

- Differentiate between system software, programming software, application software and embedded software and provide examples of languages and platforms used to create each.
- Understand basic application computing and architectural concepts, such as layered architecture types versus Service Oriented Architectures (SOA), Microservice environments, and how they have evolved over time.
- Understand consumer applications versus enterprise application environments.
- Identify software tools used most to develop apps on Windows, Android, iOS, macOS, and Linux platforms; compare and contrast the advantages and disadvantages of each.
- Describe the role of linkers, compilers, code editors, GUI designers, assemblers, debuggers, IDEs, static code analysis and code coverage tools, and performance analysis tools in developing software.

- Compare and contrast commonly used software development tools such application lifecycle management (ALM), Integrated Development Environments (IDE), Source Code Management (SCM), Test Management, Application Performance Monitoring (APM), Test Automation, Static Analysis, and other application dev and delivery toolsets. Explain how software developers ascertain and document specifications for clients, end users, or other stakeholders, and standards that software must meet.
- Discuss ways in which data can be transferred to new or updated software from existing applications or data sources.
- Explain the purpose of relational versus nonrelational database technologies.
- Describe the use of Relational Database Management Systems in creating database objects and tables, and demonstrate the ability to create a basic database in MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres or other database system.

Introduction to Cloud-Based Computing

Hours: 20–30

Sample learning objectives

- Explain the fundamentals of cloud computing and describe the challenges clients may face when transitioning to the cloud environment.
- Differentiate between Amazon (AWS), Azure, Microsoft 365 and Google Cloud Platform services, explaining the optimal uses and challenges associated with each.
- Describe the different forms of cloud computing, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and serverless.
- Explain the ways in which cloud computing services can be used to support web, app, database, mobile, analytics, networking, blockchain, development, and security services.

Programming Languages

Hours: 30–50

Sample learning objectives

- Describe the uses and functionalities of various programming languages, such as Python, Java, JavaScript, Golang, C#, C++, R, Swift, Kotlin, Ruby, etc.
- Describe the use of JavaScript in web programming and demonstrate the ability to write basic code in JavaScript.

Introduction to Software Testing

Hours: 40–50

Sample learning objectives

- State the purpose of software testing and its importance in the software development lifecycle. Provide examples of software defects that have had catastrophic effects on human health and safety, led to costly recalls and reputational damage, or resulted in major scientific or production setbacks.
- Explain the importance of domain knowledge in the software development process and describe strategies for gaining sufficient domain knowledge to guide software development and testing.
- Compare and contrast automated versus manual testing.
- Produce actionable and thorough documentation when bugs are identified, with relevant information for getting them fixed.
- Understand the role of bias and critical thinking in finding defects and failures in software.
- Describe why we design different tests like boundary value analysis, equivalence class partitioning or pairwise testing.
- Explain Agile testing and modern testing approaches and how they differ from traditional methods.
- Develop plans for testing software applications of different types and complexity and discuss the controls that will be used to ensure appropriate execution of test plans.
- Develop strategies for analyzing the design, resilience, and usability of software as part of the software test plan.
- Conduct software testing for different types of software and applications, maintaining proper records of the test protocol and results.
- Demonstrate the ability to establish and follow exit criteria, to produce summary documents of test results, and to communicate those results to others orally and in writing.
- To provide timely and useful input, describe the appropriate processes for working with software development teams based on the software development protocol being used.
- Identify the appropriate software testing libraries (packages) to use when automating test at individual phases across the software testing life cycle and justifies why and when to use them (Cucumber, JUnit, WebDriver, Cypress, Maven, etc.).

Introduction to Agile Testing and Modern Testing

Hours: 15–30

Sample learning objectives

- Describe the popular models of Agile testing and explain the pros and cons of each.
- Explain the challenges and disadvantages of using traditional waterfall testing method for testing (requirement, design, implementation, verification, and maintenance).
- Explain the V model of software engineering and testing and the benefits of starting testing at the beginning of programming and designing.
- Describe the W model of software engineering flow and testing and explain its advantages and disadvantages.
- List and describe the principles of Agile testing.
- Explain the stages of the Agile testing lifecycle and the benefits of continuous testing over more traditional models of testing.
- Describe various agile testing strategies and methodologies, explaining the advantages and disadvantages of each (i.e., test-driven development, behavior driven development, exploratory testing, extreme programming, session-based testing).
- Define Agile testing quadrants and explain how they are used to improve product development and function.
- Explain the purpose of an Agile manifesto and demonstrate the ability to create/use such a manifesto.
- Explain the importance of scrums and demonstrate the ability to form and participate in effective scrums.
- Demonstrate the ability to develop an appropriate impact assessment goal.
- Demonstrate the ability to develop an Agile test planning protocol.
- Define the term “release readiness” and explain its importance to the Agile testing model.
- Define and demonstrate the ability to perform a test agility review.
- Participate in a team engaged in Agile testing as part of a customer-driven product design.

Exploratory Testing

Hours: 15–20

Sample learning objectives

- Describe the role of exploratory testing when specific customer or performance requirements are not available.
- Explain benefits of exploratory testing when early iteration is required.
- Differentiate between and demonstrate the ability to perform unit, integration, and system testing.
- Demonstrate the ability to explore an existing application and document findings and recommendations for improvements.
- Differentiate between freestyle, strategy-based, and scenario-based exploratory testing.
- Explain the use of various testing techniques in performing strategy-based exploratory testing (i.e., risk-based analysis, boundary value analysis).
- Describe the advantages and disadvantages of exploratory testing.
- Demonstrate the ability to perform and document exploratory testing of an existing application, using each of the three testing techniques listed above.
- Explain how to differentiate between a defect or bug and a feature of an application.

Relevant military experience

Army MOS 74F and Air Force MOS 3D034 occupations include training in software testing.

Diversity, equity, and inclusion

Sixty-two percent of software testers are male, and 71 percent are white. Sixty percent of software testers are age 40 or older. Sixty-eight percent of software testers have a bachelor's degree while 14 percent have a master's degree. Only 6 percent of software testers have less than an associate degree. Expanding diversity in the software testing space is crucial to identify issues that may impact end-users from various demographics. Current efforts to diversify the software testing workforce include creating pathways for individuals without a four-year degree to pursue careers in technology (https://www.zippia.com/software-tester-jobs/demographics/).

Works Consulted

International Software Testing Qualifications Board. 2021. “Certified Tester: Foundation Level (CTFL) Syllabus.” https://astqb.org/assets/documents/CTFL-2018-Syllabus.pdf.

STATEMENT OF INDEPENDENCE

The Urban Institute strives to meet the highest standards of integrity and quality in its research and analyses and in the evidence-based policy recommendations offered by its researchers and experts. We believe that operating consistent with the values of independence, rigor, and transparency is essential to maintaining those standards. As an organization, the Urban Institute does not take positions on issues, but it does empower and support its experts in sharing their own evidence-based views and policy recommendations that have been shaped by scholarship. Funders do not determine our research findings or the insights and recommendations of our experts. Urban scholars and experts are expected to be objective and follow the evidence wherever it may lead.

URBAN INSTITUTE · ELEVATE · THE · DEBATE

500 L'Enfant Plaza SW
Washington, DC 20024

www.urban.org