# Intro

**Your workplace activity brief for Automation and Software Quality**

**As mentioned in the previous step, your first task in the Apply stage is to identify a workplace activity or project at work which you can use to apply your new knowledge and skills.**

**This step details your workplace activity brief for this module.**

By now, you should have a good grasp of some of the different techniques and tools used to test different kinds of applications.

In this Apply activity, you will **produce a written report** with your findings on:

a) investigating and evaluating the testing frameworks and methodogies your workplace uses.

b) using these tools and techniques to test applications that serve a useful purpose within your workplace. You will use your knowledge of software testing frameworks and methodologies to conduct a range of test types, such as:

* Integration
* System
* User Acceptance
* Non-Functional tests (such as Performance and Security testing)

You will be required to target your writing to address the mapped knowledge, skills, and behaviours – which are listed on the following step.

Remember, the work that you produce will be used to demonstrate evidence of your knowledge at the end-point assessment.

**Timescales**

QA expect this Apply stage to take approximately four days to complete, spread across the four weeks remaining duration of this module.

When you’re ready, select **Next** to continue.

# KSB

**Evidencing your knowledge, skills and behaviours**

**Knowledge, skills and behaviours (KSBs)** **are the core attributes you need to demonstrate in your workplace activity.**

**You must demonstrate that you are competent to excel in your profession, and to pass the end-point assessment as part of this apprenticeship.**

**Knowledge**

Knowledge refers to the learning you acquire whilst on your apprenticeship. It’s the information covered in the Discover and Practise stages. Some of the knowledge will be generic, and some will be more specific to your role. You should bear in mind the KSBs when completing the Apply stage activities, and actively demonstrate how your output covers the KSBs for this module.

* **K12**: Software testing frameworks and methodologies.

**Skills**

Skills refer to how you apply the knowledge you have gained in your job. You will build upon your skills as you progress through the modules utilising the on and off job training. There should not be a repetition of the tasks or duties that you regularly perform.

* **S5**: Conduct a range of test types, such as Integration, System, User Acceptance, Non-Functional, Performance and Security testing.
* **S13**: Follow testing frameworks and methodologies.

**Behaviours**

Behaviours mean the way in which you work. You need to act in a way that is appropriate for your organisation and profession.

* **B4**: Works collaboratively with a wide range of people in different roles, internally and externally, with a positive attitude to inclusion and diversity.
* **B10**: Committed to continued professional development.

# Format and Structure

**Structure**

The following is a checklist of what should be included in your final submission.

**Introduction**

* Description of the project: Explain what you will be discussing in the case study (i.e., the software solution being developed).
* The approach.
* Project outcomes.
* How the KSBs are evidenced through your workplace activity.
* How the project was assigned to you.
* The stakeholders you liaised with for this.
* The key requirements.
* Other considerations e.g., budget.
* The way you interacted with various stakeholders.

Remember to write in first person and clearly state your personal input.

**Part One: Testing Frameworks and Methodologies**

Your task in this first part is to:

* investigate the testing frameworks and methodologies your workplace uses.
* evaluate the use of various software testing frameworks and methodologies and how would you justify their choice.
* add these findings into your portfolio as evidence of the frameworks used and the difference types of testing performed at your company.

**Part Two: Application Testing**

Your task for the second part is to:

* design test scenarios to test a complex application that serves a useful purpose within your workplace.
* analyse and design the tests you have chosen to write, thinking about:
* what you wish to test and why.
* methods which you use to test the application (manual or automated testing methods).
* the test environment and how it can be setup by another employee who may be unfamiliar with your tests.
* the use of different categories of tests such as Unit, Integration, and system testing where it is necessary.
* categorising these tests further into functional and non-functional tests.

You may pick a substantial segment of an application which you use at work.

Plan your time carefully to ensure your work is of a high quality. Consider the following questions:

* What are you testing? (Which part of the app AUT or CUT)
* Why are you doing it? (User requirement or other reasons)
* Who is the test designed for?
* Where is the work taking place? (Test setup environment)
* When are the tests taking place? (When should the test be run and how often?)
* How is the work planned and managed?

The areas highlighted below will provide a good foundation:

1. Analyse what you intend to test. Make sure each test can be referred to system or user requirement.
2. Create different categories of tests. For example, Unit, Integration, System testing.
3. Further categorise your tests into Functional and Non-functional tests.
4. Create a set of test data. Make sure each test targets a single functionality of the application. Use a standard template used by your company if one exists.
5. Identify and record the expected and actual results.
6. Write code for each test in your chosen programming language. You may choose any language, but we suggest using C#, Java, Python or JavaScript.
   * Identify the code's purpose, by tracing it back to the functionality being tested.
   * You may use IDs for each requirement for tracing purposes.
7. Create user documentation for other technical staff to enable them to configure the test environment and run your tests.
8. Store all of the code changes you have produced using version control. Using Git would be an example of version control software that you can evidence.
9. Ask a colleague to review your code changes. Provide evidence of this (e.g., screenshots of comments). An example could be comments on the cloud based repository known as GitHub.

**Sensitive information**

You may be working in sensitive or restrictive environments, which means that you cannot share certain information. In cases like this, you can anonymise anything you complete.

It is also important for you to recognise where you should be redacting information when sharing evidence with third parties. This demonstrates good practice in security, and compliance with your company’s security policies.

**STARRS**

When providing evidence during your apprenticeship journey, it is recommended that you use the STARRS method for each work task you are describing. This will provide you with a clear direction on how to structure the required evidence into the portfolio.

* **Reflect:**
  + What did you do well?
  + What didn’t go as well as you had hoped?
* **Result**:
  + What did you do after you completed your actions?
  + Did anything need updating?
  + Did you communicate with anyone?
  + What was the conclusion?
* **Task:**
  + What were you tasked with doing?
  + What were your objectives?
  + What did you do, and what did the task involve?
  + Explain your tasks and responsibilities.
* **Situation**:
  + What is the situation?
  + What is the issue/problem that needs solving?
* **Action**:
  + What did you do?
  + How did you do it?
  + Who did you communicate with?
  + What tools did you use? (This is where you can provide your screenshots/evidence.)
* **Strengthen**:
  + In later stages, when you look to consolidate your portfolio, have you provided the best evidence from your apprenticeship work?
  + Do you need to collate additional supporting evidence?

# Submit your final copy

The following list is a reminder and checklist on what you should include in your final submission.

(Remember, you will need to have had your summary signed off by your DLC before completing your workplace activity, and writing and uploading your final submission.)

* **Introduction**
  + Description of the project.
  + The approach.
  + Project outcomes.
  + How the KSBs are evidenced through your workplace activity.
  + How the project was assigned to you.
  + The stakeholders you liaised with for this.
  + The key requirements.
  + Other considerations e.g., budget.
  + The way you interacted with various stakeholders.
* **Part One: Testing Frameworks and Methodologies**
  + investigate the testing frameworks and methodologies your workplace uses.
  + evaluate the use of various software testing frameworks and methodologies and how would you justify their choice.
* **Part Two: Application Testing**
  + design test scenarios to test a complex application that serves a useful purpose within your workplace.
  + analyse and design the tests you have chosen to write.