Title: Test Automation Engineer

Aim: Learners will obtain a comprehensive and practical understanding of test automation. They will gain the mindset and skills that empower them to make informed decisions on automation strategy through implementing automation.

Target Learner: Individuals who are looking to start building or expanding existing skills in automation.

Pre-requisites: Experience in Software Testing in a professional environment. Have worked in a software development team. Basic knowledge of coding would be advantageous. Able to describe what testing and automation are within the context of software development.

Core Competencies: Coding, Modelling, Communication, Personal Professional Development, Sharing knowledge, Knowledge of what automation is, Oracles, Heuristics, and Regular self-reflection. Note: core competencies are a set of proficiencies a learner will develop throughout the curriculum.

Foundation Level			
Module (Task)	**Aim of the module**	**Steps (Task steps/core competencies to cover)**	**Learning Outcomes** By the end of the curriculum, the learner will be able to:
Evaluate tool options	This module covers competencies around the identification, analysis and decision making process of choosing what automated tools to work with in a given context.	Identify tool requirements	● List requirements to consider when building or selecting a tool for automation ● Plan out the requirements for a required tool

		Research which tools can and cannot be used	● List different sources of information for tool research ● Carry out research to discover tool options ● Prioritise tool options to decide which tooling to use
		Create an example of tool delivering value	● Recognise the value of creating a prototype with a real example ● Carry out a timeboxed session to evaluate the suitability of a tool
		Evaluate the value of the tool created or selected	● List different criteria to determine a tool's suitability ● Reflect on a timeboxed prototype to determine a tool's suitability
		Sharing knowledge of tooling with others	● Share experience of evaluating a tool's suitability with others
Create a minimum viable automated feedback loop	This module covers competencies that enable an individual to create initial automated checks in a framework that demonstrates the value of automation approach and highlights future tasks to carry out.	Setup framework with required libraries and configurations	● Outline the basic features of an automation framework ● Choose what libraries to use as part of an MVP automation framework ● Implement necessary libraries and configuration to create an MVP automation framework
		Create a space to version control the framework in	● Describe how version control works ● Use version control tools to create a space to store an MVP automation framework

		Create an initial check to demonstrate necessary abstractions	● Relate to how establishing an initial check helps us create an MVP automation framework ● Outline the steps required to build an initial automation check ● Create an initial automation check for an MVP automation framework ● Request feedback on framework through code reviews and discussions
		Integrate framework into a build pipeline	● Outline how automated checks can be part of a build pipeline ● Detect where in a build pipeline an MVP automation framework can be added ● Integrate an MVP automation framework into a build pipeline
		Setup feedback from a build pipeline	● Relate to why reporting from automated checks is an essential part of the feedback loop ● List information that could be valuable to report from automation frameworks ● Share reports from an MVP automation framework with the team / stakeholders
		Create documentation for relevant details on a framework	● List necessary information to add to framework documentation ● Create relevant documentation for framework

MINISTRY OF TESTING

		Sharing knowledge of framework with others	● Share a conclusion on creating and using an MVP automation framework
Investigate Failed Automated Checks	This module covers competencies around reacting to when automated checks fails and identifying and executing actions to resolve issues that arise from failing automated checks.	Gather and analyse failure information	● List ways in which a build might have failed ● Review errors in a build to uncover the problem
		Determine the cause of the failed check	● Outline different reasons why an automated check might have failed ● Detect why an automated check has failed
		Determine what to do with a failed check	● Outline steps to take to solve different failures from automated checks ● Determine steps to take to resolve a failing automated check ● Implement a solution to resolve a failing automated check
		Explore issue highlighted by failed check	● Determine what area of the system requires exploring based on failure feedback ● Run an exploration of a system to learn more about the area of a system that a failed check has highlighted

		Communicate details of failure to the team	● Gather the necessary details to share with a team on a automated check's failure ● Share with the team why an automated check failed and what to do
Developing or adopting Testing Tools	This module covers competencies around analysing where automated checks will be executed, identifying requirements to allow execution to occur and setting up automation execution runs.	Research potential tools for future use	● List different sources of information to learn about new tools ● Choose tools to learn about ● Execute short sessions to learn about new tools
		Implement a basic working solution to use during testing activities	● Identify problems in which tools can help with ● Evaluate whether to use a tool or build one ● Implement a tool to create a basic working solution to a problem
		Evaluate success/issues with new tooling and plan next steps	● Consider whether the tool has helped to solve a basic problem ● Formulate a plan to continue solving a testing problem with tools
		Educate others about how to use tooling	● List ways to communicate details about a new tool to others ● Share with others how to use new tooling

Maintaining Testing Tools	This module covers competencies around supporting and enhancing tooling that is used to support testing activities beyond automated checking.	Evaluate if tooling is still providing value	● List ways to measure if a tool is still delivering value ● Question if a tool still delivers value ● Conclude whether to continue using a tool under evaluation
		Expand tool features based on testing requirements	● Determine what new features or steps are required for an existing tool ● Modify tooling to expand its features
		Keep tooling up to date	● Relate to why keeping tools up to date is important ● Review if a tool requires updating
		Maintain documentation for tooling	● Assess whether documentation for a tool is out of date ● Update documentation for tooling

Intermediate Level

Module (Task)	**Aim of the module**	**Steps (Task steps/core competencies to cover)**	**Learning Outcomes** By the end of the curriculum, the learner will be able to:
Creating Automated Checks	This module covers competencies to identify what automated checks to create, what system layers to implement them against and build said automated checks.	Decide what automated checks to create	● Break down system behaviour to understand how features are implemented ● Point out risks that might impact different parts of an implementation ● Prioritise risks to create automated checks for

		Determine what layer the automated check should be on	● Determine if testability allows tests to be automated on an identified system layer ● Decide what layer to create automated checks on
		List what state, actions and assertions the check will carry out	● Outline the anatomy of an automated check ● Describe different types of state to set up for an automated check ● Describe how automated checks can interact with a system under test ● Describe ways in which assertions can be codified in an automated check ● Decide what state, actions and assertions need to be created for an automated check
		Codify the creation of state	● Evaluate what tools need to be used to create state for an automated check ● Develop implementation for creating state
		Codify the actions the automated check will take	● Evaluate what tools need to be used to interact with a system under test ● Develop implementation for interacting with a system under test
		Codify the oracle in the automated check (assertion)	● Evaluate what tools need to be used to assert a system under test ● Develop implementation for asserting a system under test

		Run the automated check against the product to debug issues	● Run automated checks to confirm they are running correctly
		Have automated checks reviewed by others	● Outline ways in which code reviews can improve automated checks ● Carry out code reviews on created automated checks
Maintaining Automated Checks	This module covers competencies around understanding why maintenance of automated checks is required and how to carrying out maintenance to improve checks and reduce flakiness.	Listen and react to situations that indicate maintenance is required	● List reasons why maintenance might be required for automation ● Determine what maintenance steps are required for automated checks ● Implement changes to improve automated checks
		Evaluate checks to make sure they are still adding value and delete/update ones that don't	● Describe why deleting automated checks is sometimes required ● Articulate different characteristics that can be used to judge automated checks ● Critique existing automated checks ● Modify or delete automated checks that no longer deliver value
		Maintain checks that appear to be flakey	● List reasons that cause automated checks to be flakey ● Determine reasons why an automated check is flakey ● Resolve issues that contribute towards an automated checks flakiness

Reporting Automation Results	This module covers competencies around analysing what automation results are valuable to stakeholders and implementing tooling to collate and share results with said stakeholders.	Collect and store useful information as the automation is running (Screenshots, HAR files 2, log files etc.)	● List different sources of information that can be useful to report automation results ● Determine what technical information stakeholders want from an automation report ● Choose which tools to use to collect and store information ● Use tools to collect and store useful information
		Collate results from automation and send results to stakeholders via tooling (Email, dashboard, Slack, test case tools)	● Choose which tool to use to present or alert stakeholders of results ● Implement tooling to send results to a team
		Review automation reports regularly	● Review automation reports to determine future actions
Maintaining a Framework	This module covers competencies around understanding what maintenance a framework might require as well identifying and executing maintenance steps.	Listen and react to situations that indicate maintenance is required	● List different ways that a framework might need maintaining ● Assess whether a system requires maintenance
		Maintaining dependencies and keeping tooling up to date	● Outline why updating dependencies and tools is important ● Describe issues that might occur during dependency update ● Identify dependencies to update ● Update dependencies and resolve issues in updates for a framework

		Maintaining good coding practices	● Identify what are good coding practices for frameworks ● Explain to with team/stakeholders which coding practices to adopt ● Use coding practices to maintain and improve a framework
		Maintaining documentation for framework	● Assess whether documentation for a framework is out of date ● Update documentation for a framework
Manage execution of automated checks	This module covers competencies around analysing where automated checks will be executed, identifying requirements to allow execution to occur and setting up automation execution runs.	Setup an environment for executing automated checks	● Describe the pros and cons of different environments to run automated checks in ● Select an environment to run automated checks in ● Build an environment to run automated checks in
		Modify an environment for automated check execution	● List ways in which an environment might need to be configured for automation ● Break down tasks to prepare an environment for automation ● Modify an environment for automated checks
		Identify what automated checks to run and when	● Describe the value of running automated checks in specific categories ● Organise automated checks into categories and run them in a set order

		Arrange triggers for running automated checks	● List different ways in which automated check runs can be triggered ● Judge which type of trigger to implement for automated check runs ● Implement triggers for automated check runs
Advanced Level			
Module (Task)	**Aim of the module**	**Steps (Task steps/core competencies to cover)**	**Learning Outcomes** By the end of the curriculum, the learner will be able to:
Creating the automation strategy	This module covers competencies around analysing a project's context, modelling systems and using what is learnt to propose a strategy that will ultimately be adopted by stakeholders.	Model a given context	● Relate to why context matters when making choices in automation ● Gather information about a context ● Create a model of a context
		Measure the testability of a given context	● Describe what testability is ● List sources of testability ● Use existing models to guide testability analysis ● Analyse a context to determine its testability
		Identify and set automation goals	● Describe what an automation goal looks like ● Align automation goals with testing strategy goals ● Consider how automation might help in a given context ● Formulate a list of automation goals

		Document the strategy	● Identify a strategy's audience and their needs to aid communication ● Consider different approaches to documenting a strategy ● Document an automation strategy
		Seek buy-in and sign off on strategy	● Relate to why communicating a strategy is important to it’s adoption ● Relate to why you want feedback on your strategy ● Experiment with influencing techniques to improve the adoption of an automation strategy ● Request feedback on an automation strategy ● Reflect on feedback for an automation strategy ● Update an automation strategy based on feedback
		Maintain the strategy	● Recognise strategies go out of date ● List reasons why a strategy changes ● Survey a given context to identify new changes ● Recommend changes to an automation strategy ● Update an automation strategy on a regular basis

Planning to automate	This module covers competencies around identifying, arranging and executing plans that will help a team achieve a strategies goals.	Determine what plan to create	● Use different techniques to discover explicit automation opportunities ● Consider what options are available to achieve a strategy's goals ● Choose an option to help achieve strategy goals
		Document an automation plan	● Relate an automation plan to an automation strategy ● List the different attributes of an automation plan ● Construct an automation plan
		Get feedback on an automation plan from team	● Relate to how feedback from a team can help improve an automation plan ● Collect feedback on an automation plan from team members
		Slice up tasks to carry out	● Break down the tasks to execute an automation plan ● Create tasks for each step of an automation plan
		Evaluate progress and success of an automation plan	● Conclude whether the automation plan has been successfully delivered ● Consider changes required to update an automation plan

MINISTRY OF TESTING

Contributors

Aaron Flynn
Abhishek R Nair
Abolade Fadairo
Adeel Mansoor
Amit Paunikar
Andie Johnson
Andrew Clark
Andrew Kelly
Armando Cifuentes González
Artem Stupakov
Arthur Shevchenko
Atara
Atmaram Naik
Ben Poudrier
Beth Marshall
Bhadmus
Bob Salmon
Brian King
Butch Mayhew

Caleb Crandall
Chaissy Daniel
Charles Penn
Chris Marquis
Christine Barbato
Christopher S. Holmes
Christopher Scott Holmes
Clay
Colin Sullivan
Conrad Braam
Constance Armitage
Courtney Johnston von Nieda
Dan Leighton
Daniel Cleaton
Daniel Erasmus
Daniel Hunt
Dave Harlowe
David Adams
David Maynard
David McGill

MINISTRY OF TESTING

David Muñoz Gomez
Deb Sherwood
Deborah Reid
Dejan Pupinoski
Del Dewar
Djsksl
Dustin Altermann
Eamon Droko
Ed Manlove
Edgardo Crovetto
Elizabeth Zagroba
Emiola Ibironke
Erin Donnelly
Fidelis Okafor
Gabbi Trotter
George Katsaros
Glenn
Gregory Paciga
Han Lim
Honey Chawla

Hylke de Jong
Ifat Sharifi
Irja Straus
Isabel Evans
Ish Abbi
Jack The Lantern
Jan Reimann
Jaswanth Manigundan
Jay Smith
Jesper Ottosen
João Farias
Joe DeMeyer
Jokarr Morris
Jon Blackburn
Jon Lane
Jonathan Terry
Kamaxi
Katja Meyer
Kevin Stechler
Kirsty Ireland

Konstantinos Koukoumpetsos
Kristof Van Kriekingen
Kyle Potter
Landon Montgomery
Larry G
Lauren Cousin
Lee Burlow
Listya Widyasari
Louise Gibbs
Maaike van der Linde-Haisma
Makedonka Andeva
Manuel Fidalgo Sicilia
marcela oliveira
Mario Specht
Mark Winteringham
Martin Gijsen
Matt Charlton
Matt Drinkwater
Matthew Bretten
Mike Talks

Mirza Sisic
Nebojsa Petkovic
Neoklis Dimakos
Niall Carolan
Niall Crawford
Nicola Lindgren
Nikolina Djekic
Oleg Grudko
Oleksandr Romanov
Paul Naranja
Payal Juneja
Peak
Peet Michielsen
Penny Howard
Peter Anderson
Rashmi
Remco van Bree
Richard Bradshaw
Richard Spurr
Rishi Davda

MINISTRY OF TESTING

Rob Backaller
Robin McKenzie
Sam Connelly
Sam Hodson
Sandeep Singh Thukral
Sandra K
Saransh Vaid
Sean Redikin
Sebastian Byrum
Sebastian Stautz
Sergejs Patriks Vosels
Shona McClarence
Shreya Agrawal
Shuk Tak Woo
Shweta Sharma
SinRedKon
Stefan Bollmann
Stephanie Bogart
Stuart Crocker
Subha

Sue
Sugi
Sukma Ragil
Swaraj Kumar Barik
Talha Ozleblebici
Tamoya Beckford
Terence R.
Thorsten Prehn
Tsveta Krusteva
vidhya Arumugam
Vitalii S.
Xander Bartels
Zachary Lysobey
Zoe Jobson