TESTER SKILLS

CONTENTS

TESTING CORE

FUNDAMENTALS

- Tester mindset
  - I know how to explain a testing mindset.
  - I know the difference between a testing mindset and dev mindset.
- What is QA?
  - I understand the meaning of quality assurance.
  - I understand that being a QA alone cannot assure quality.
- Testing approaches
  - White box, Gray box, Black box, Test oracles, Test prioritization
  - I know when to use which approach when.
  - I know the pros and cons of each approach.

TESTING TECHNIQUES

- User acceptance, Exploratory, Sanity, Regression, Smoke, Unit, Integration
- I know when to use which techniques when.
- I know the pros and cons of each technique.
- I understand in which order these techniques should be applied.

TEST DESIGN AND EXECUTION

- Tests analysis
- Strategy, Planning
- Mind mapping
- Exhaustive testing
- Risk based

EXPLORATORY TESTING

- Heuristics, Session based, Test charters, Persona based
- I know how to apply different exploratory testing techniques.
- I know when it is valuable to do more structured testing.

TEST REPORTING

- I can create test plans
- I can create test cases
- I can track test case progress
- I can build a dashboard

DEFECT MANAGEMENT

- I know how to log a defect.
- I know when to log a defect.
- I know how to classify a defect.
- I know the difference between all the defect types.

TIME MANAGEMENT

- I know how to prioritize my time when under constraints
- I know how to focus my efforts
- I know how to limit context switching

CONTEXT DRIVEN TESTING

- I understand the impact of quality software on the clients.
- I know the criteria for a risk incident.

LEARNING RESOURCES

INTRODUCTION TO TESTING

- Open Lecture by James Bach on Software Testing
- Agile Software Testing with James Bach
- Black Box Software Testing (BBST)
    - Foundations
    - Bug Advocacy
    - Test Design
    - Exploratory Testing
- Create Your Tester Portfolio, Issi Hazan & Shmuel Gershon
- How to get started in Software Testing, Tim Elbe, Adventures in QA
- Four Hour Tester, Joep Schuurkes and Helena Jeret-Mae
- Whiteboard Testing Videos

THE TESTING MINDSET (WITH SOME LOGIC AND PHILOSOPHY)

- List of Cognitive Biases, Wikipedia
- Propositional Logic, Kevin deLaPlante
- Agile Testing Mindset and the Role of the Agile Tester, Amir Gharai, Testing Excellence
- Philosophy for Beginners, Marianne Talbot, University of Oxford CCE
- General Philosophy, Peter Millican, University of Oxford

TEST DESIGN AND EXECUTION

- Requirements Analysis, University of Michigan
- Test Design Techniques (Videos)
- Heuristic Test Strategy Model, James Bach
- An Introduction to Scenario Testing (Cem Kaner)
- EuroSTAR Software Testing Video: Ten Minute Test Plan with James Whittaker
- CAST 2014 Keynote - Test Cases are Not Testing: Toward a Performance Culture
- The Art and Science of Questioning, Jessica Ingrassellino
- Software Testing Best Practices, Rex Black
- Test Matrices, Michael Bolton

TESTING TOOLS AND TECHNIQUES

- What Testing Tools should I Learn, Mark Winteringham, Ministry of Testing
- Investing in Testing - The Importance of the Right Technique, Rex Black
- Top Test Management Tools

EXPLORATORY TESTING

- What is Exploratory Testing, James Bach
- Rigorous Exploratory Testing, Elizabeth Hendrickson
- Exploratory Testing, gov.uk Service Manual
- Evolving Understanding About Exploratory Testing, Michael Bolton
- Testing and Noticing, Michael Bolton and James Bach
- Google Tech Talks - Exploratory Testing, Jon Bach

- Test Heuristics Cheat Sheet, Elizabeth Hendrickson
- Exploratory Testing, Martin Mudge
- Session Based Testing, Peter Tennekes
- Session Based Test Management, Jon Bach

TEST REPORTING

- Test Execution Reports, SoftwareTestingHelp
- Effective Test Status Reporting, Rex Black

DEFECT LIFECYCLE MANAGEMENT

- Bug Advocacy - How to Win Friends, Impress Programmers and Stomp Bugs, Cem Kaner
- Defect Life Cycle: Essentials of Software Testing, Jennifer Marsh, Udemy Blog

DEVELOPING AS A TESTER

- 30 Days of Testing, Ministry of Testing
- Buccaneer-Tester: Winning Your Reputation
- Becoming a World-Class Tester, Ilari Henrik Aegerter
- The power of believing that you can improve, Carol Dweck, TedTalks

SDLC

AGILE

- scrum, kanban, xp
- I know the different agile models
- I know when to apply different the models
- I understand how to avoid mini waterfall, wagile, etc.
- Story breakdown
- Shift left
- Agreed standards
- Whole team culture
- Designing for testability
- Postmortems

TEST DRIVEN DEVELOPMENT (TDD)

- I understand the meaning of TDD.
- I know how to implement TDD.
- I understand the impact of TDD on the agile process.
- I know how to drive a TDD culture.

LEARNING RESOURCES

AGILE

- Awesome Agile
- What is Agile?
- Agile in a nutshell
- Agile Software blogs by Mike Cohn
- Agile Software Guide - Martin Fowler
- What is Scrum?
- Agile 101
- Agile Manifesto
- Agile Methodology Tutorial for Beginners
- Planning Poker
- Agile Retrospectives
- Agile Retrospective Meetings
- Agile Sprint planning
- 7 Lessons Agile Can Teach Us about Leadership
- DZone Agile blogs
- Agile Testing - LambdaTest Learning Hub
- Agilemania's blogs

TESTING IN AGILE

- Agile Manifesto Principles
- Agile Testing Overview, Elizabeth Hendrickson
- Agile Test Planning with the Agile Testing Quadrants, Lisa Crispin and Janet Gregory
- Quick Tools for Agile Testing, Lisa Crispin and Janet Gregory

TDD LEARNING MATERIALS

- Awesome TDD
- TestDrivenDevelopment - Martin Fowler
- TDD Is The Best Design Technique - Dave Farley
- The 3 Types of Unit Test in TDD • Dave Farley • GOTO 2022
- Test Driven Development vs Behavior Driven Development • Dave Farley • GOTO 2022
- TDD | Dave Farley's Blog
- Test *Driven* Development - Dave Farley
- Test Driven Development is the best thing that has happened to software design
- TDD Quick Guide

LEADERSHIP / MENTORING AND COACHING

LEADERSHIP

- Values
    - Leads by example
    - Can influence
    - Is accountable
    - Shows up with high energy
    - Is actively engaged
    - Can execute on demand
- Good communication, presentation, facilitation & interviewing skills

MENTORING AND COACHING

- I understand the difference between mentoring and training
- I understand how to encourage other testers
- I understand how to coach a testing mindset
- I understand how to help a tester keep engaged

RECRUITMENT

- I can confidently evaluate potential candidates.
- I can assess the testing level of a candidate
- I can confidently review a candidates CV.
- I can confidently review a candidates technical assessment.
- I can assess cultural fit to the organization.
- I can confidently converse with a candidate.

LEARNING RESOURCES

LEADERSHIP

- https://theqalead.com/quality-engineering-planning-strategy/leadership-in-test-test-planning/

NON-FUNCTIONAL TESTING

LOAD AND PERFORMANCE TESTING

- Benchmarking
- Stress testing
- Tools
    - k6
    - lighthouse

SECURITY TESTING

- Vulnerability scanning
- OWASP top 10
- Secrets management

FUNCTIONAL SQL PERFORMANCE TESTING

- query-optimization
    - optimizing-joins, reducing-subqueries, selective-projection
- query-analysis
    - I can test one SQL script against another one to see which one runs more efficiently.

LEARNING RESOURCES

LOAD AND PERFORMANCE TESTING

TUTORIALS

- A Complete Performance Testing Guide With Examples
- What is Performance Testing?
- Master Performance Testing
- Performance Testing vs. Load Testing vs. Stress Testing
- What is Throughput in Performance Testing?
- Load Testing Best Practices
- Tools and Techniques for Performance and Load Testing
- awesome-performance-testing
- 15 Top Load Testing Tools Open Source MUST KNOW in 2021
- 15 BEST Performance Testing Tools (Load Testing Tools) In 2023
- Performance Testing Tools: Types, Use & More

K6

- Beginner's Guide to Load Testing with k6
- API performance testing with k6
- K6 - Usage Guide and Docs
- K6 - Examples and Tutorials
- K6 - API Load Testing
- How to get started with K6?
- Grafana K6
- Performance Testing with K6

GOOGLE LIGHTHOUSE

- Lighthouse - GitHub
- What Is Google Lighthouse and How to Use It?
- Introduction to Chrome Lighthouse

YOUTUBE CHANNELS

- Actionable Performance Testing & SRE Tips on the TestGuild Performance Podcast
- K6
- JMeter Tutorials
- How To Test Software Performance
- What Is Google Lighthouse and How to Use It?

SECURITY TESTING

TUTORIALS

- OWASP Top 10
- ZAProxy
- Security Testing Guide
- Security Testing Tutorial
- OWASP Web Security Testing Guide
- Web Application Security Testing Guide
- Security Testing
- OWASP Mobile Security
- Hacksplaining
- OWASP Top 10 Vulnerabilities
- OWASP Top 10 2021
- OWASP Top 10 tools and tactics
- awesome-security
- awesome web security
- awesome Hacking
- awesome web hacking
- awesome mobile security
- awesome-appsec
- awesome-api-security
- Threat Modelling
- What is Threat Modelling and How does it work?
- What is CVE?

PENETRATION TESTING

- Penetration testing guide
- Pentesterlabs
- The Beginner's Guide to API Hacking - DANA EPP'S BLOG
- Hacking APIs: Workshop - Corey Ball
- MalAPI by mrd0x
- What is DAST?
- What is RAST?
- What is SAST
- What is IAST
- Difference between DAST,SAST, IAST and RAST
- SAST, DAST and IAST
- CyberSecurity - Codeacademy

- Collection of Penetration Testing resources
- Penetration Testing Full Course - FreeCodeCamp.org
- APISec University - API Penetration Testing Free Course
- APISec University - API Security Fundamentals
- APISec University - OWASP API Security Top 10 and Beyond!
- APISec University - API Security for PCI Compliance

YOUTUBE CHANNELS

- OWASP Foundation
- OWASP Top 10 2021 - The List and How You Should Use It
- 2021 OWASP Top Ten Overview - F5 DevCentral
- API hacking for the Actually Pretty Inexperienced hacker with Katie Paxton-Fear
- OWASP Devslop
- A Starters Guide to Pentesting with OWASP
- Actionable Security Testing Tips on the TestGuild Security Testing

SECURITY TESTING TOOLS

- Vulnerability Scanning Tools
- 19 Powerful Penetration Testing tools
- Top 10 Open Source Security Testing Tools for Web Apps
- 10 best mobile app security testing tools
- API Security Tools
- Snyk.io
- Pynt.io
- Black Duck
- Checkmarx

SQL TESTING

SQL BASICS

- What are relational databases?

DATA TYPES

- I understand the common different data types that SQL uses (e.g. VARCHAR, DATE, DATETIME, BIT, NUMERIC, INT, CHAR)

DATA DEFINITIONS

- create-table, alter-table, truncate-table
- data-constraints
    - primary-key, foreign-key, unique, not-null, check

DATA MANIPULATION

- select
    - I can select all rows and columns from a table
    - WHERE clause.
    - ORDER BY.
    - I can use a SELECT DISTINCT
- insert
- update
- delete
- join-queries
    - inner-join, left-join, right-join, full-outer-join, self-join, cross-join
    - I understand the difference between an inner join and a left/right join and the use-case for each.
    - I can select a subset of rows and all columns from multiple tables.
- aggregate-queries
    - sum, count, group-by, having

ADVANCED SQL FUNCTIONS

- strings
    - concat, length, substring, replace, upper, lower
    - I can select a substring of data from a specific column of data.
    - I can use REPLACE to swap out specific strings within data points.
- date-time
    - date, time, datepart, dateadd, timestamp
- numeric
    - floor, mod, abs
    - I can perform arithmetic on specific numeric values (SUM, AVG, MIN, MAX etc.) and return these values.
- conditional
    - nullif, case, coalesce
- sub-queries
    - types
        - scalar, column, row, table

    - I can write a select query within my primary select query
    - I can select a subset of columns from a table by specifying the specific columns that I want returned.
- nested subqueries, correlated subqueries, recursive queries, pivot unpivot, common table expressions (ctes), dynamic SQL
- window functions
    - row_number, rank, dense rank, lead, lag

STORED PROCEDURES

- I can turn an existing stored procedure into a select query to debug.
- I can create a basic stored procedure that takes one or more inputs.
- I understand the difference between a table, a view and a stored procedure.

- Views
    - creating, modifying, dropping
- Temp tables and variables
    - I can declare variables of different data types and use them inside my SQL query.
    - I can create a temporary table within my SQL query.
    - I understand the difference between a temp table and a table variable and know when/how to use them.

LEARNING RESOURCES

DATABASE TESTING

- Database Testing Complete Guide
- Database Testing Tutorial - Softwaretestingmaterial.com
- Database Testing Tutorial - Tutorialspoint
- Database Testing - Javatpoint
- Introduction to Database Testing

LEARNING MATERIALS

- What is a Database?
- Different Types of Databases - JavatPoint
- Database types - Mongodb.com
- Types of Databases - Tutorialspoint
- DBMC and SQL basics
- Database Management System

SQL/MYSQL TUTORIALS

- SQL Tutorial - SQLTutorial.org
- SQL Tutorial - W3Schools
- SQL Tutorial - SQLZoo.net
- SQL Tutorial - Tutorialspoint
- SQL Tutorial - Javatpoint
- SQL Tutorial - Programiz
- SQL Tutorial - Full Database Course for Beginners
- SQL Cheat Sheet
- Visualize your SQL queries
- MySQL tutorial
- MySQL Tutorial - W3Schools
- MySQL Tutorial - Tutorialspoint
- Learn to use MySQL Database - FreeCodeCamp.org
- MySQL Workbench

APPS

ACCEPTANCE/COMPONENT/MOCKED TESTS

- I understand why we do app acceptance tests
- I can write an app test
- I can update screenshot snapshots
- I can find elements using CSS selectors, ids, custom attributes
- I understand where to find puppeteer commands
- I can create fixtures
- I can debug a test
- I understand the test architecture/ framework
- I can setup the framework on an app repo
- I can debug in CI

VERIFICATION TESTS

- I understand why we do app smoke tests
- I can write a test
- I can debug / troubleshoot a test locally and CI
- I understand what a good smoke test is

PLAYWRIGHT / PUPPETEER

- I can find the documentation
- I understand the relationship of alpine, chrome and pptr
- I can emulate different screen sizes
- I can inject jQuery when needed
- I can find the documentation
- I can find and interact with elements
- I understand the relationship of docker, browsers and the package

FRONTEND FUNDAMENTALS

- html-css-javascript
- browser-devtools
- ajax
- caching
- spa
- csr-vs-ssr
- responsive-vs-adaptive

REACT

- I can build an existing app
- I can update selectors with qa tags
- I understand why they are called SPAs
- I understand the relationship of app and bff
- I can review an app unit test
- I understand the distributable

LEARNING RESOURCES

PLAYWRIGHT

- Playwright Documentation - Nodejs
- Playwright Documentation - Java
- Playwright Documentation - Python
- Playwright Documentation - .NET
- Playwright - YouTube
- Playwright Locators
- Playwright with JavaScript
- What is Microsoft Playwright JS?
- Testing Modern Web Apps with Playwright | OD110
- An End To End Playwright Testing with TypeScript
- Playwright Tutorial: Getting Started With Playwright Framework
- Playwright - Github
- Playing with Playwright
- Interactive website to learn playwright

APIS

API VERIFICATION TESTS

- I can execute tests locally
- I can execute tests against DEV|INT|UAT
- I can group tests
- I understand the naming convention for new test files / folders
- I can determine unnecessary tests
- I can determine which tests should be acceptance tests
- I can use lib test data to replace hard coded variables

API ACCEPTANCE/COMPONENT/MOCKED TESTS

- I understand the purpose of mocked tests
- I can execute these test locally
- I can observe the mocked requests
- I can find the downstream api dependencies
- I can create a fixture manually
- I understand mock server matching algorithm
- I understand the naming convention
- I can update my fixtures so they are unique as possible
- I can group my fixture for different scenarios
- I can manage fixtures for reuse

SCALA/ GOLANG

- I can create a docker image for an API
- I understand what the build command means
- I can find the built docker image
- I can override the image with a pre built image

LEARNING RESOURCES

INTRODUCTION TO API TESTING

- REST API Concepts and examples
- Testing Strategies in a Microservice Architecture
- What is an API? API for Beginners
- What is API Testing?
- API Testing Tutorial
- A Comprehensive API Testing Guide
- Getting Started With Testing Microservices
- JSON.org
- What is JSON?
- APISec University - API Documentation Best Practices
- APISec University - API Tools and Resources

CI/CD/CT

MONITORING / ALERTING

- Kibana
- Grafana
- Pager duty
- Dynatrace

KUBENETES

- Pods vs containers
- Deployment files
- Resource management
- Scaling
- Deployment patterns
- KubeCTL commands
- Orchestration
    - Clusters
- I can create this deployment setup
- I can add in config maps
- I can add in secrets
- I understand purpose of each .yml file
- I can update the Individual .yml files
- I can use common variables
- I can create specific deployment files for different environments

HELM CHARTS

- I can create this deployment setup
- I can add in config maps
- I can add in secrets
- I can create a values file
- I understand how this setup is different from k8s

RANCHER

- I understand clusters, services, pods, workloads, secrets, config maps
- I can find a service
- I can restart a service
- I can filter a namespace
- I can delete a namespace
- I can find secrets, config maps values
- I can find logs
- I can run shell commands

LEARNING RESOURCES

LEARNING MATERIALS

- Free Devops Books
- Lets-DevOps/awesome-learning
- What is CI/CD?
- What is CI/CD Pipeline?
- annfelix/DEVOPS-WORLD
- Continuous Integration Patterns and Anti-Patterns
- Learn How to Set Up a CI/CD Pipeline From Scratch
- How to use Docker for Automation Testing?
- Test Automation in DevOps
- Continuous Testing with Azure DevOps
- Scaling tests with Docker
- Whole Team Approach to Continuous Testing

BLOGS

- DevOps.com
- ContinuousDelivery.com
- Continuous Deliver in DevOps
- Devops Testing
- Practical Guide to Continuous Intergration for Automation Testing
- Jez Humble's Blog
- Dave Farley's Blog
- Gasper Vitta's Blog(Lots of cool stuff on CI/CD, docker, testing)
- What is Continuous Testing?
- Spacelift.io

ASYNC

RABBIT MQ

- I can publish a message
- I can check for consumers
- I understand what exchanges, queues and routing keys are
- I understand what is a queue, exchange, consumer, event, command
- I can check a local queue
- I can add in missing rabbit config

CAMUNDA

- I can start a workflow
- I can pause a workflow node
- I can locate a workflow incident
- I can migrate workflows between versions
- I can claim a human task
- I can execute a human task
- I can cancel a workflow in batches
- I can utilize the cockpit to monitor processes
- I can update variables in Batches
- I can modify variables
- I can troubleshoot a stuck workflow
- I can correlate a message
- I can locate an active workflow
- I can retry nodes
- I can navigate to called process definitions
- I can check access rights.

LEARNING RESOURCES

RABBITMQ

- https://www.rabbitmq.com/getstarted.html
- https://www.classcentral.com/subject/rabbitmq
- https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html

REPORTS / EMAILS / SMS TESTING

STATIC DOCUMENTS

- I can test readability and comprehensibility
- I can test the layout and design
- I can test the functionality, the accuracy and consistency
- I can do visual regression on a document
- I can test bulk generation
- I know how to use a mocked sms and email servers
- I understand whitelisting email addresses and cell numbers
- I understand how template engines work
- I know how to block dynamic parts of a document for visual regression

LEARNING RESOURCES

PDF TESTING

- https://preflight.com/blog/the-top-10-strategies-for-effective-and-efficient-pdf-testing/

LOCAL TESTING

DOCKER

- Containers
    - What and why we need them?
- Installation
- Persisting data
- Using 3rd party base images
- Building images
- Dockerfile
- Container registry
- CLI commands
    - Images
    - Containers
    - Volumes
    - Networks
- I can build and image
- I understand public and local images
- I can create, delete, manage image versions
- I can install additional apps on an image
- I can install certificates as an image

DOCKER-COMPOSE

- I can create a docker-compose file
- I can create a docker-compose override file
- I can add a service
- I can create volumes and working directories
- I can expose ports
- I can create dependent services
- I can create health checks

VS CODE

- I can navigate a project
- I can install extensions
- I can setup auto linting

VM SETUP

- I can set my BIOS for virtualization
- I can find the latest VM image
- I can install all the tools I need on the VM
- I can setup my SSH keys for GIT
- I can configure virtual box
- I can update the installation ssh scripts as needed
- I can install Linux apps
- I can update my guest additions

DOCKER COMPOSE

- Introduction to Docker Compose
- Docker Compose Documentation
- docker/compose
- Parallel Execution of Tests using Selenium Grid 4 with Docker Compose

DOCKER

- Docker
- Play with Docker
- Learn Docker Online
- Docker Curriculum
- Learning Docker
- veggiemonk/awesome-docker
- The Ultimate Guide to End to End Tests with Selenium and Docker
- Docker full Course
- Training - Play with Docker
- Docker Tutorial for Beginners by Mosh
- Docker Labs
- Docker Tutorial for Beginners | Docker Full Course
- Docker Tutorial for Beginners - A Full DevOps Course on How to Run Applications in Containers
- Docker Containers and Kubernetes Fundamentals – Full Hands-On Course
- Docker Tutorial for Beginners [FULL COURSE in 3 Hours]
- Docker Tutorial for Beginners – Introduction & Getting Started

VS CODE

- https://code.visualstudio.com/learn
- https://learn.microsoft.com/en-us/training/modules/introduction-to-visual-studio-code/
- https://code.visualstudio.com/docs/introvideos/basics
- https://code.visualstudio.com/docs

TOOLS AND TECH

GIT

- I can clone, add, commit, push
- I understand branches vs master
- I can resolve merge conflicts
- I understand the naming conventions branches and commits

GITLAB

- I can create an MR
- I can review an MR
- I understand all jobs in the pipeline
- I can kick off a pipeline
- I understand the gitlab.yml file
- I can find a failed job
- I can setup a schedule

JEST

- I can write a test
- I can group a test suite
- I understand different "expects" available
- I understand snapshots
- I can setup jest framework
- I understand different cli commands available
- I understand different config values available
- I understand different test hooks available

JAVASCRIPT

- I understand ES6
- I understand classes
- I understand different data types (arrays and objects etc..)
- I understand file and variable naming conventions

NPM

- I can update multiple packages when needed
- I know how to check for outdated packages
- I know the process for updating a package
- I know the difference between npm I and npm ci
- I know where node modules come from
- I know where node modules are cached
- I understand the purpose of node modules
- I understand what the package.lock file does