



SOFTWARE TESTING ACADEMY

# NATIONAL REPORTS ON EXPERTS QUESTIONNAIRS

This project has been funded with support from the European Commission.  
This report reflects the views only of the author, and the Commission  
cannot be held responsible for any use which may be made of the  
information contained therein



# INDEX

<b>1. DEMOGRAPHIC</b>	<b>3</b>
<b>2. HOW LONG HAVE THE EXPERTS BEEN WORKING AS SOFTWARE TESTERS?</b>	<b>3</b>
<b>3. WHAT ARE THE FIELDS OF EXPERTISE OF THE EXPERTS?</b>	<b>3</b>
<b>4. WHAT DO EXPERTS THINK ARE THE MOST IMPORTANT KEY SKILLS OF SOFTWARE TESTING PROFESSIONALS?</b>	<b>3</b>
<b>5. WHAT SHOULD PEOPLE WHO ARE INTERESTED TO WORK IN SOFTWARE TESTING LEARN?</b>	<b>3</b>
<b>6. HOW WOULD THE EXPERTS STRUCTURE A LEARNING PATHWAY REGARDING SOFTWARE TESTING?</b>	<b>3</b>
<b>7. ARE THERE KEY MATERIALS THAT A SOFTWARE-TESTING ACADEMY SHOULD INCLUDE?</b>	<b>3</b>
<b>8. ARE THERE KEY MATERIALS THAT A TEST AUTOMATION ACADEMY SHOULD INCLUDE?</b>	<b>3</b>
<b>9. IS THERE SPECIFIC KNOWLEDGE THAT THE SOFTWARE TESTER NEEDS TO KNOW?</b>	<b>4</b>
<b>10. IS THERE SPECIFIC KNOWLEDGE THAT SOMEONE WHO IS WORKING IN TEST AUTOMATION NEEDS TO KNOW?</b>	<b>4</b>
<b>11. IS THERE SPECIFIC KNOWLEDGE THAT THE SOFTWARE TESTER NEEDS TO KNOW REGARDING ACCESSIBILITY TESTING?</b>	<b>4</b>
<b>12. MAJOR CONCLUSIONS</b>	<b>4</b>



# 1. Demographic

## Gender

Female: 8

Male: 12

Diverse: 1

No answer: 3

**Age:** 23 – 51 **average:** 39,18

## Job

- Test consultant → 4
- Test Automation Engineer → 2
- Test manager
- Technical Product Owner
- Team Lead
- Quality Engineer → 4
- Senior Product Manager
- Lead IT Quality Engineer
- Principal Tester
- Software Engineer
- Software Test Consultant
- Web developer
- Analyst
- Integration testing engineer
- Developer
- Tester in functional testing or automated

## Degree of education

- High school – 3
- Masters – 11
- Bachelors – 5
- PhD – 1



- Others – 3
- Keine Antwort – 1

## 2. How long have the experts been working as software testers?

From 0 to 22 years, 7,38 years in average.

## 3. What are the fields of expertise of the experts?

Test automation, test management, test process improvement, manual testing

E-commerce, user testing, user experience testing, leadership and analysis

Integration testing, Software development, Code analysis

Hardware programming, Industrial Automation, Web programming, Microcontrollers

Test analysis, test and data design, test strategy, session based testing, test consulting, exploratory testing

## 4. What do experts think are the most important key skills of software testing professionals?

- Curiosity
- Experience and technical knowledge
- Communication skills and team player/empathy/cooperation
- Analytical thinking and skills
- Critical thinking and awareness for quality
- Open minded
- Creativity/thinking out of the box/lateral thinking



- Ability and interest to learn/ learning skills
- Skills in automation
- Accuracy
- Understanding the code
- Concentration
- Organization
- Discipline
- Attention to detail, knowledge of web workers and Microsoft Office. As an attitude, it is quite positive that the person is proactive, does not get paralyzed by bugs in the tested software and has a somewhat inquisitive attitude to get results, as well as a certain curiosity about how things work, in addition to patience.
- Flexible
- basic English
- Tenacity
- Agility
- speed of adaptation to change
- nonconformity
- versatility
- Resilience
- Mastery of different programming languages, mastery of the processes of a software, knowledge of frequent errors and critical points in the processes and services of a program.
- Debug, Strong Technical intuition, programming and hardware skills.
- Systemic and analytical vision; knowledge of the client's business; methodology.
- Flexibility to learn new languages.

## 5. What should people who are interested to work in software testing learn?

- Testcase design techniques
- Exploratory testing
- Writing Bug Reports



- Review techniques (Code, Requirements or other Testcases)
- Software development 101 and Software delivery lifecycle
- The costs of bugs and risk assessment
- Principles of black box testing, test automation academy, blogs by known people in the industry.
- Reaching out and asking for guidance and mentorship - many testing experts are willing to learn.
- Heuristics of testing
- Basic programming
- Tools: Jira, Confluence, Excel, depending on working field: Postman, Cypress, Java, Javascript, Gherkin, how to work with developer tools
- Software architecture
- Automation of testing
- Understanding end users
- Fluency in using software tools
- Both the theoretical and the practical part should be taken into account for testing.
- Learning covering many types of situations.
- Learning concepts of manual testing, automatic testing, white and black box testing.
- Learning to think, to question things, to stay sober when the whole team is upset.
- Learning software development life cycle and testing principles.
- Learning to function in the different types of testing, the possible processes involved and the products or environments to be tested.
- Learning about bug reporting, exploratory testing and programming, at least at a basic level.
- Learning to work in a state of uncertainty.
- Learning to use the tools necessary for your job, good organisation and good habits. You will also learn the necessary English, a breakdown of functional documents and how to obtain them, and how to work well in a team.
- Learning what testing means and why it should be done, together with basic testing techniques depending on what they are going to do.



- Group dynamics, various automation tests, etc.
- Use different OS and, in the automatization case, Selenium and Python
- Programming languages
- Programming, Debugging Tools
- Systems analysis
- Algorithmic

## 6. How would the experts structure a learning pathway regarding software testing?

- ISTQB basics → agile testing techniques → specialization in either test management or test automation
- What is Software → How is Software Developed → Why do we need to test this software? → How do we test this software? (e.g. test design techniques, exploratory testing) → Something doesn't work as expected, what do we do? (analysing the failing testcase, writing a bug report if necessary) → how can we improve in software testing (e.g. starting early in the process by reviewing requirements or finding "bugs" in refinements)
- Books (e.g. The art of software testing, "How to test" by Mike Talks), Internet blogs and knowledge bases, Experiments on internet sites/shops (find the bug sessions), Courses provided by trusted companies
- Start with a combination of dry theory (test techniques) and practice (find SUT to practice on, devise strategies, test, write bug reports etc).
- start with learning the basic vocabulary, techniques for functional and non functional testing, ways of working - i.e. agile and waterfall, some automation and infrastructure
- Read books, listen to podcasts, go to conferences, listen to experts and think critically. Question everything!
- Download frameworks and do experimentation
- Learn coding basic and concepts and HTML, CSS, HTTP, API, Web-stuff
- Find a senior quality engineer and work together



- Manual testing → using test automation
- Theory of testing linked with knowledge of software → work under supervision of somebody more experienced
- From basics to more advanced topics in using software tools
- First, provide general information on how to structure a test case, how to report an incident or failure, how to classify low, medium and high impact, and how to report the results.
- The ideal would be to start with the basics of testing and go from the simple to the complicated, in the case of automation the main issue is to have knowledge of languages, mainly HTML
- Basic 'blackbox testing' certification (ISTQB or BBST), learning manual test design, programming language (java or js, etc.), automation framework (selenium, or cypress, etc.), knowledge of protocols and basic architectures.
- First teaching the basics, then exploratory testing, and finally automated testing.
- Focusing on the software development lifecycle and how the QA team should intervene in each phase
- First functional and then to automate.
- The best thing to do would be trial and error, trying out many ways and models until trainees find the one that works best for them.
- Basic testing theory, test case analysis, test case writing, test execution, bug reporting.
- Know what test runs are, know how test runs work, understand what test runs are, learn with someone the fundamentals of test runs, learn to carry out unit tests together with a programmer and thus making the leap to operational testing.
- First selenium-web, then appium and then different frameworks such as playwright
- Objectives, who is a software tester, programming languages, test methods, automated test procedures, critical points in a program.
- Knowledge of operating system, programming, debugging.
- It would depend on the software.



## 7. Are there key materials that a software-testing academy should include?

- Gregory, J., & Crispin, L. (2019). Agile Testing Condensed: A Brief Introduction.
- Hendrickson, E. (2013). Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing
- ISQTB Foundation Level with agile Extension
- <https://theartfultester.com/blog/2018/07/06/what-is-software-testing/>
- BBST materials by Cem Kraner <https://docs.ast-bbst.org/>
- Ministry of Testing Dojo <https://www.ministryoftesting.com/>
- The Scrum Tester (book), The Game Of Testing
- to build software and exercises
- Session based testing, exploratory testing, test charta, test design methodology, 3 amigos, good bug reporting, test pyramid, test process (classic vs. agile),
- Automation frameworks
- Some API testing concepts, some UI testing concepts, techniques for root cause analysis, observability concepts, performance and security testing concepts
- Rapid Testing, Context Driven Testing, DevOps Fundamentals
- Problem solving methods
- Thinking in systems, risk analysis
- Exploratory Testing - Elisabeth Hedrickson
- Rapid Software Testing - Michael Bolton
- Nothing specific
- Certification path
- Videos
- Test pages with bugs and content adapted to a testing academy.
- Basic certification 'blackbox testing', manual test design learning
- In addition to the theory, notions about Regex, Infrastructure and programming
- Subject provided by the ISTQB



- Understanding of test types and brief introduction to jira, confluence and testrail.
- Bug reporting, exploratory testing, creating test plans, creating test cases.
- Teaching people to be autonomous and to solve problems without blocking themselves.
- How to work with a repository, how to have a group dynamic, how to deal with problems arising from testing, manual testing and automated testing.
- SIM - Operating System, debug
- Programming; Systems analysis

## 8. Are there key materials that a test automation academy should include?

- <https://testautomationu.applitools.com/>
- <https://clean-code-developer.de/>
- Something about architecture of test automation is often forgotten. Do not focus on one automation language.
- Panther ODM on github
- No specific materials
- Materials related to Selenium
- I do not know
- SELENIUM and others
- Practices based on the theoretical classes, it can be the development of a small project (Analyze a solution, automate and report results / executions)
- The programming language that you will use in addition to HTML and a good page with a bug.
- Basic knowledge in programming language (java or js, etc.), automation framework (selenium, or cypress, etc.)
- Java and/or Python
- Git or github, intellij, brief explanation about how jenkins works, of course, learning about the framework and java, as it is often used.



- Programming basics and then specializing in the own FW or the one necessary for the project
- Teach how to deal with the frustration of selectors that are not in the DOM.
- Basic materials of the programming language to be used (java, javascript, python...), documentation and examples of the main testing libraries (selenium, codeceptjs...), inclusion of cloud computing services (aws)
- How to work with a repository, how to have a group dynamic, how to deal with problems arising from testing, manual testing and automated testing, links to the frameworks doc and simple installation guides for Appium, etc.
- None specified any materials.

## **9. Is there specific knowledge that the software tester needs to know?**

- Depends highly on the project/work context
- Not only testing but also constructive techniques (how to prevent bugs) and organisational techniques (how to increase quality with organisational changes). Otherwise the test only fights the symptoms but not the root cause
- Testing techniques, testing approaches, how to prepare testing documentation, how to document work, how to raise issues are only examples
- product specific knowledge that one can acquire while working on that specific project
- To test well you need to know what is it for, how that helps to solve the problem
- As for related topics of usability, domain specific aspects - those is wise to add when testers knows the product abd it's purpose well enough.



- How to communicate the value they add, how to assess, prioritise and mitigate risk and communicate this effectively to stakeholders/devs
- Basic coding and CS concepts
- Using the specific tools
- Proper understanding the types of the testing
- No
- ISTQB Theory
- Basic technical knowledge and communication skills
- That the person is somewhat curious, has initiative and the basics of computer science.
- Basic programming
- The importance of communication, abstraction, basic usability knowledge.
- To communicate the results of the tests carried out. Complete flow of the testing phase.
- One respondent answered that there is no specific knowledge that a software tester needs to know, but a proactive attitude and curiosity are more than desirable before entering academia. In my opinion, a tester is made and born. As for technical knowledge, it can be acquired without problems when the tester has the right attitude and his personality invites him to do it.
- Working with Jira, agile methodologies and office in general. Manage your own expectations and those of other teams. I think that in addition to using their work tools, everything depends on the project and the client. It is essential to know what the client needs and to acquire knowledge about the product. A good tester must know how to seek knowledge.
- Some programming language and testing tools. Using different OS in an agile way.
- Algorithmics.
- SIM - Programming Languages.
- Software business area.
- Knowledge of the language.



## 10. Is there specific knowledge that someone who is working in test automation needs to know?

- at least one programming language
- Programming skills, pair programming, common programming languages
- Software Architecture, Design Patterns, DevOps
- Good manual testing
- How to prepare test data and how to choose test cases for automation
- Selenium tools
- Global knowledge of how the most popular frameworks work.
- Basic HTML and the language to use.
- Key aspects in test maintenance: synchronisation, data management, locators. Good test design.
- Planning so as not to lose your way when automating.
- Technical knowledge in REST and SOAP services.
- It is quite important to have at least a minimum knowledge of HTML because it shortens learning times.
- Basic programming language, java for example. Selenium, Cypress, Karate, Gherkin, Cucumber, X-Ray, Jira, Java, Jenkins, Python, Testrail
- Depending on the complexity of the product, in-depth knowledge of the programming language to be used may be required.
- Programming languages that you use in your day-to-day work and testing tools.
- Programming the object type of the page, API operation and getting selectors from the DOM.
- Algorithm.
- YES - Debug
- Programming
- Knowledge of PLCs and / or microcontrollers



## 11. Is there specific knowledge that the software tester needs to know regarding accessibility testing?

- WCAG guidelines and considerations to make around these
- testing techniques with screen readers, knowledge of UI elements like ARIA labels which support accessibility
- a good understanding of different kinds of disability and the difficulties these can create for end users if we don't make accommodations for them.
- Understanding that improved accessibility improves experience for all users not just those with disabilities
- You cannot test accessibility as a non-handicapped person properly.
- Always include a person the specific needs
- Proper understanding of all types of tests, including accessibility
- Standards related to accessibility
- Types of disabilities and how each of them impede the software experience
- Usability and security related knowledge.
- Knowledge related to usability and security.
- Must be able to put oneself in the shoes of others.
- Create accessibility tests from scratch several times with different methods. WCAG checklists
- Important aspects in terms of inclusivity for better UX
- Know accessibility standards so that testing meets the standards and know how to put yourself in the shoes of people with special needs so that testing goes beyond and maximises accessibility (or know how to research the topic).
- According to one respondent, it would depend on the product. In most cases it would not be necessary, but it could always be an added value, especially when working with graphical interfaces.
- Generic knowledge of disabilities and particular needs.
- YES - Special resources.
- Constraints and possible critical scenarios.



- Algorithm.

## 12. Major conclusions

1. A lot of skills and specific knowledge depends on the project, workframe, customer etc.
2. Curiosity and thriving for learning new things are very important as well as social skills and analytical thinking
3. Experts are always willing to learn. It is necessary to stay open minded even as an expert and to keep your knowledge up to date.
4. Mid age males with Masters degree dominate the sample group
5. Testing experts come from different IT fields, especially software development
6. Qualities such as analytical thinking and accuracy as well as knowledge of software are crucial.
7. Testing professionals should understand both users and the software itself
8. For experts, people interested in software testing should learn from the basics to the more complicated.
9. The contents that are most important among expert testers - apart from knowing the theory, which is a necessity - are knowledge about programming languages, especially HTML and Java, as well as Selenium for automatization testing.
10. The most important skills to have by testers are proactivity or attention to details or organization.