BCSHCI 2024 IwC Journal Special Issue, 2025, pp. 1–19

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

The Software Testing Community and IT Stereotypes: A Study with Industry Professionals

Isabel Evans,[1,*] Chris Porter[1] and Mark Micallef[2]

[1]Dept. of Computer Information Systems, Faculty of ICT, University of Malta, Msida, Malta and [2]Dept. of Computer Science, Faculty of ICT, University of Malta, Msida, Malta
[*]Corresponding author. isabel.evans.17@um.edu.mt (Mobile: +44 (0) 746 7 122 067)
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Testing is essential for successful delivery of software development and maintenance cycles, and is performed by specialist testers, developers, and other team members. Our earlier studies of testers found a wide diversity in the participants. The intention for this study was to understand more about the characteristics, backgrounds, and work experiences of testers, as well as the approaches they take to the activities of testing, with the research question 'Who is testing?' In a qualitative survey of over 70 industry participants, covering testers from multiple countries and domains, the study uncovered information about their wide range of backgrounds, hobbies, roles and characteristics, with differing work styles and problem-solving preferences. The people contributing to testing during software projects have varied backgrounds, academic qualifications, hobbies and interests. Examination of their job titles and aspirations showed the actual and potential scope of the role. Their responsibilities, approaches to testing activities, and the problems they described showed their work requires a high cognitive skill level. We contribute findings that testers do not meet the stereotypes for IT workers, and the role does not meet the stereotype of boring, repetitive work. This matters for tester recruitment, retention, and career paths. There are also implications for the representation within IT teams of people using software, and potentially for society. Breaking the stereotyping and supporting diversity in testers' backgrounds and characteristics might be supported by using tester personas to support aspects of testers' work life. We demonstrate that a simple set of personas would not reflect the rich heterogeneity of the software tester community, and instead introduce our current work to build a framework of heuristics that aids test tool designers and those acquiring tools to address building personas for their context.

Key words: Empirical studies in HCI — Programming teams — Software Testing — People in Testing — Diversity

Introduction

Software testing is an activity that provides essential information during software projects, and is performed not only by specialist testers and developers, but by other project stakeholders, team members and customers. The purpose of testing includes providing information about risk, quality, defects and fitness for purpose [32, 34]. Our earlier studies of software testing, for example [19, 20], indicated a wider range of people performing testing than expected, and this was also indicated by [24]. Our findings included a wide range of educational backgrounds and previous experiences. This motivated the research question **'Who is testing?'**. A qualitative study of over 70 testers provides empirical evidence of testers' diverse backgrounds and preferences. Providing an HCI lens on one group in the IT industry allows investigation of potential personas among people who test software. Open questions in an anonymous survey allowed them to share stories of their backgrounds, roles, responsibilities and interests.

Taking an interpretivist approach [6] allowed indicative patterns to be uncovered about the reported backgrounds and experiences of people testing software. These patterns can inform the understanding of the testing community and potentially improve recruitment, onboarding, training, work practices and tool support for testers.

Using an HCI lens, it has been possible to understand more about testers and their roles compared with other IT workers [39, 59]. Stereotyping is challenged by finding a wider range of people in testing than the IT recruitment databases suggest, and by embracing this diversity as a benefit. This paper studies the diversity of background, qualification, work preferences, and interests of software testers, rather than diversity of race, ethnicity and disability [51]. Whilst the latter is important to society and individuals, the literature we examined showed stereotypes for IT workers around education, interests and communication, for example, [39, 22, 23, 3]. While the stereotype for testing is a boring and repetitive role, with conscientiousness as the chief required characteristics, we find testers have multi-faceted, cognitively challenging, communication-heavy roles; therefore, the recruitment and support of testers need to encourage diversity of problem solving and those who represent the end users of software. Breaking stereotypes and encouraging heterogeneity enables fresh perspectives, especially for problem solving, and better testing of software and systems [4, 15, 53], and we show this is supported by a wide range of entrants to testing, not primarily computer science graduates [3, 36]. This affects hiring practices, team dynamics, and team composition, as well as work patterns and the design of tools to support testing. We found that the range of interests and educational background, including arts and social science, supported a range of communication styles and work styles. We contribute a greater understanding of software tester community building on and adding to the work of others such as [24]. We also contribute an understanding that the ambition in earlier work [19, 20] to overcome a narrow view of usability by test tools designers with a set of tester personas was not possible; this study demonstrated the difficulty of building a simple set of tester personas that encapsulate the rich variety in the community. Based on the research findings, we have started to operationalise the information about testers' characteristics into a framework of heuristics, idea-t, which, along with the data on the heterogeneous range of testers, is our contribution.

The next section of the paper will describe the background to the work, grounded in HCI and Software Testing. The research methods and results are then described. The discussion explains why taking a people-centric HCI lens to the testers' stories matters for society, for building a thriving test community, and for better work practices and tools

Background and Related Work

The participants for this study are software testers, sometimes identified as from a software engineering background [3] and focused on automating tests [23], but often with wider backgrounds, remits and interests [14, 47, 19]. These testers are not working in HCI, UX or usability, although some of them might take part in activities in those areas. The focus of this study is to use HCI methods to better understand these software testers. This section discusses testing, the activities of the study participants, and HCI, which provides methods and concepts for the study.

Testing

Software testing is an essential part of any software development or maintenance project. It is heavily relied on by teams and organisations to provide information about the readiness of software for its customers. Yet it has been seen as difficult and expensive [13]; the need to reduce costs while increasing quality and certainty remains a driver for C-level decision making [14]. Definitions of testing may be divided into human-focused, creative and cognitive [5], and the older, more mechanical, such as the earlier work of [7]. The former lend themselves to testing organised as people-centric with tool support, while the latter take a tools-centric, automation-driven approach. The purposes of testing include providing information about software quality and risk, uncovering defects and identifying conformance to requirements. Both [5] and [32] mention test activities including planning, preparation, test execution, results analysis, risk analysis, and reviewing artefacts, while [10, 9] both show the rich variety of activities and exploratory approaches that a software tester might take. These activities, in contrast to HCI testing and evaluation, generally do not involve observing users, although they can do. This is because the user interaction with the software is only part of what is being tested. The testing might range from working with the technology and infrastructure levels, with no user interface, or in contrast, engagement with societal risk and ethical concerns at an organisational level, and system to component testing between those.

Over the history of the IT industry, software testing has been both a specialist role, and an activity that is part of a team-wide responsibility [38, 58]. Over the last few years, with a move to DevOps, the emphasis has been to integrate testing into a whole-team activity; for example, the 2022 State of Testing Report stated: *'In many organizations, other team members also take part in the formal testing process. For example, Developers, Product Owners, Support, End Users . . . '* [47]. Testing is not the reserve of specialist testers and developers; instead, many people are doing some testing as part of their role. The latest World Quality Report [14] focuses on the quality engineering role as a new testing role, with communication across organisations and up to C-level, and preventative work, as well as automation and adoption of AI tools to aid both efficiency and effectiveness of testing. It is now an activity done by many people within a team.

Although the IT industry has moved towards the reduction of testing headcount by attempts to automate testing and by merging testing activities into other roles, this does not remove the need for expertise in testing. Reports contemporary with the data collection for this study (2021-2022), such as [38, 47, 13, 23] show specialist testing skills are still required, as well as domain, business and UX testing skills. Indeed, the 2021 World Quality Report from [13], whose respondents are selected from senior managers to C-level executives, remarks *'Testing efficiency is always top of people's minds [sic]' and also notes that it is 'both interesting and striking . . . that the most important factor [for productivity] was deemed to be having enough staff with the right skills.'* The aspirations of C-level managers include having specialist quality and testing roles as champions and experts on quality, customer experience, and productivity remain key in the 2024 World Quality Report [14]; role titles moving to Quality Engineers, including preventative and customer/business facing work, requiring both an understanding of business and good communications across the business. This contrasts with the view of testing as a destructive activity of bug finding, and requiring mainly technical, computer science skills, as seen by, for example, [37, 8, 3, 23].

While [61] showed that practitioners who are not already testers may view testing as an undesirable career, perceiving the work as easy but tedious, they also found that people already engaged in software testing careers talked about the challenging, interesting nature of the work. The practitioners in [47] and [13] view testing as a challenging job: *'a tester was and continues to be the equivalent of a Swiss army knife for his or her team. This means that we need to have a combination of skills that will serve our needs depending on the challenge being thrown at us. . . . flexibility, critical thinking, and the ability to learn all the time are the most necessary tools any tester needs to have to succeed in our trade.'* [47, page 22], yet only 3% of the recruiters in [22] have critical thinking as a key skill. The 2024 State of Testing Report [48] shows that communication continues to be an important skill, with critical thinking, and continuous professional development needed to support increasingly complexity. Exploring what it means to be an expert in software design, [46] used as one of their markers of an expert that 'experts test'. Of course, that does not mean that every tester is an expert, but it is an indicator of the importance of testing expertise in the software development process.

HCI

The study of Human-Computer Interaction (HCI) and the practice of User Experience (UX) design and testing are closely related; both focus on a better understanding of people's contexts, characteristics, goals and needs [45]. Applying those techniques to software testers could be both useful and instructive. Potential benefits include a better understanding of actual and potential diversity among testers, informing recruitment, onboarding, work practices and tool design. Usability defined as the effectiveness, efficiency and satisfaction of a user carrying out a task with specified goals [33] is part of the goal of HCI, which also includes other product quality attributes and qualities in use that affect the person's experience; for software testers this is discussed in our papers [20, 19], which identified the need for a better understanding of tester personas, to aid tool design and the testers' experience of those tools. In this paper, our focus is to build on the earlier work to gain a better understanding of the software testers.

In this study, software testers are the participants and their concerns and narrative are the focus of the research. HCI techniques have been applied to software testing before, for example by [11, 12], to assess task difficulty and cognitive workload when testing, and by [24, 37] to examine skills and characteristics of testers. Investigation of other roles in IT, for example, [26, 35, 55, 60] used HCI methods to investigate developers' experiences with their tool sets. They looked at visual environments, static analysis tools, the effect of warning and error messages from tools, and the impact of push notifications.

HCI uses both qualitative and quantitative methods, including in-depth investigation of small sample sizes, listening to and observing the participants, and focusing on solutions to people's challenges. It uses an experimental or an exploratory approach [49] and results are reported either as metrics or as qualitative data [2, 52]. In this work, mixed methods are used, using an exploratory approach, and reporting both on metrics and qualitative data, providing empirical results (see Method section). The key HCI component for this research is the focus on the Human rather than the Computer, the idea of user-centred design [50], and understanding whether personas could be built for testers and their tools.

Personas and Stereotypes

HCI uses people-centred, qualitative methods to understand both the context and goals of those who will use a specific software tool, identifying their relevant characteristics that affect the software's design. One such method is persona development, a rigorous empirically-driven process, which enables designers to reason about and increase their empathy for, those who will be using their product or system, and take design decisions, based on these 'fictional characters' [17]. Personas tend to be developed for a specific product, and are developed as named individuals.

Personas are built from data collected with techniques such as interviews with actual or potential users, or may be derived from statistical data about system usage if the intent is only to understand existing users. The research to understand and maintain a set of personas for a system can be a project in itself; [1] remarks that while ad-hoc personas might be developed in an afternoon, for a rigorous persona one might take several months. As a compromise, they suggest one to two weeks as a reasonable length of time for persona development. There is a sliding scale of rigour, from a fully researched persona set, to assumption and ad-hoc persona developed in less than a day, to finally stereotypes which contain misunderstandings, bias and prejudice at their core. This is a trade-off between time spent and the accuracy of the information gained about the potential people who will use the product.

Although personas are intended for product design and design testing, they may also be used in other activities, for example, to inform the design of testing tools, or to guide recruitment, onboarding, and career development. However, if that is done, care is needed to ensure that an attempt at categorisation does not become stereotyping, with its inherent bias. Studies by [3, 23, 36] indicate a potential bias in understanding personas for testers to focus on computer scientists, for example, including the assumptions for recruitment and onboarding, perhaps disadvantaging candidates from outside computer science. As discussed in Background Section on testing, candidates outside computer science may be increasingly needed to meet C-level requirements. In this study, understanding how personas emerge from testers' backgrounds, interests and communication styles might support a wider range of backgrounds and their preferred approaches to their training and work could reduce stereotyping.

Stereotyping and Diversity

HCI practices increasingly emphasise the need to study and understand a diversity of potential and actual people who will be using the software [30]. In an article on testing AI, [53] suggests that diversity in teams brings benefits: 'muddy boots pragmatists' looking for risks, problems and solutions to reduce harm and remove bias must be part of the design process. Their article concludes 'The diverse workers of this camp ... have important messages to ensure that the blue-sky dreams can be channeled into realizable products and services that benefit people and preserve the environment.' In [15], the authors describe how, given the wide range of backgrounds, personalities and preferences of people using software, a different or wider range of people may be needed for software testing. If C-level managers need to recruit testers with both the right attitudes and the potential to rise to the organisation's aspirations for quality, then recruiting a widely diverse, interesting group of people to these challenging roles is important to the success of software development and delivery.

Despite the challenging nature of testing, as described by industry practitioners and experts, testing can be perceived as boring or repetitive [61]. This perception by non-testers of the work being boring is not an incentive to join the testing industry. To recruit testers, and increase diversity, the job needs to be shown as interesting, with opportunities for growth. In [59], the authors find that boring people are perceived as both emotionally cold and often as less competent. They note that certain jobs and hobbies are seen as boring, and therefore, the people who do those jobs are assumed to be boring. In [39], the authors examine stereotypes as expressed in the O*NET, a mechanism for assessing role suitability by reference to hobbies and interests. O*NET classifies people according to set categories of interests: Realistic, Investigative, Artistic, Social, Enterprising and Conventional (RIASEC). Their study sampled 1000 actual and aspirant computer science workers and used the RIASEC scale to assess their interests. In O*NET the typical computer science worker is expected to score low for social, artistic and multiple interests, and [39] suggests that this stereotype blocks entry from more diverse groups. Studies examining factors for tester effectiveness and personality traits [36, 37] suggest particular personality traits, which, while possibly effective if testing is seen as essentially a destructive, bug-finding task, are less applicable to the modern tester as quality engineer [14, 48].

While researchers such as [22, 23] examined the skills requested by recruiters, in this study, we asked participants about their activities in work and their interests as a way to understand testers, and compare them both with with the O*NET stereotype for people suitable for recruitment into IT, and also compare testers with the findings of [39] for IT people in general.

Expertise and Experience

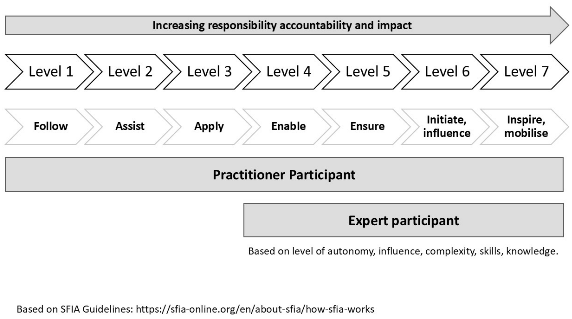

Fig. 1. SFIA Levels against Participant Practitioner and Expert Status

Expertise is gained partly through years of experience, and also by training and continual professional development. The Skills Framework for the Information Age (SFIA) [25] has seven levels, from 'follow' to 'inspire and mobilise' (Figure 1). In SFIA, expertise is based on: the autonomy in which someone works; their level of influence; the complexity of the work they do; their skill level; and their knowledge. Continuous professional development is expected. For example, the membership grades for the British Computer Society[1] require a relevant academic degree or equivalent to become a member (MBCS), with additional skills and responsibilities to become a Chartered IT Professional (CITP), based on the SFIA. To become a Fellow of the BCS (FBCS), further progress through the seven SFIA levels are required.

This relates to Nemeth's levels of skill [43] compared to SFIA in Table 1. Work is rule-based, skill-based or knowledge-based, and the mix of those work types will change across the SFIA levels. Clearly, certain aspects of testing are more rule-based, and other aspects are knowledge-based. Similarly, in SFIA, level 5-7, by the nature of the responsibilities, are less likely to be rule-based and more likely to be knowledge-based. In the SFIA role descriptions, testing and programming are set at levels 1 to 6, with specialities such as penetration testing from level 3 to level 6. No specific quality engineer role is shown in SFIA, but the nearest equivalents, Quality Management and Software Life Cycle Management, both range from level 3 to level 7. A practitioner at level 1 to 3, working rule-based and following directions, is not someone who can influence and inspire improvement, change and the implementation of preventative measures across an organisation. Quality engineers as described in the latest World Quality Report [14] are clearly working at the higher end of the SFIA levels, indicating that 'testing' moves beyond the mundane and rule-based.

An interview study of 19 senior professionals by [24], finds that testers without computer science (CS) degrees would prefer to gain testing experience as a module in degrees other than CS, and training courses for all new employees in testing, while generally their participants favoured learning through doing. This range of training and experience opportunity supports testers from multiple backgrounds entering the profession, or contributing to testing as part of their role, in a structured way.

[1] See `https://www.bcs.org/membership-and-registrations/become-a-member/professional-member-criteria/` (Accessed 26 February 2025)

Method

A mixed methods approach is used, providing empirical results from qualitative data.

Survey Design and Piloting

The design of the anonymous online questionnaire included a mix of open and multiple-choice questions, based on the themes from literature such as [19, 20, 35, 39] was used to identify relevant questions, quotes, codes and themes to help with question design and later, data analysis.

The survey was presented to the participants in three sections, with an informed consent at the start and a chance to drop out after each section, confirming whether the responses up to that point were permitted to be used in the research. The three sections were:

- Participants' backgrounds including demographics, education, qualifications, jobs before their current role, and hobbies;
- Participants' current roles, years of experience in IT, job description, additional responsibilities, and how closely their job title matched their actual work;
- Participants' activities to carry out their roles, the approaches, techniques and tools they used in their work.

The draft survey was reviewed with other researchers, and then piloted. The pilot participants came from different countries, with different levels of experience in testing, and included native English speakers and those with English as a second language. The survey questions were refined based on their feedback. The survey questions are available in the Appendix.

Running the Survey

The survey was released as a Google Forms questionnaire, open for responses from September 2021 to July 2022, during which time it was publicised via online postings, industry conference announcements and word of mouth. This provided both convenience and snowball sampling to start the survey, as the link to the survey was shared through networks, with participation propagating beyond the researchers' network. The

Role	Examples	Skill Type	SFIA Equivalent
decision maker	manager	knowledge-based	SFIA level 5-7
monitor (vigilant or supervisory)	lifeguard or airline pilot	rule-based	SFIA level 3-5
information processor	clerk	skill-based	SFIA level 2-4
closed loop controller	machine operator	skill-based	SFIA level 1-3
information encoder and storer	trainee	skill-based	SFIA level 1-3
discriminator, pattern recogniser	interpreter	knowledge-based	SFIA level 3-7
ingenious problem solver	scientist, designer	knowledge-based	SFIA level 5-7

Table 1. Human Roles in Systems (after Nemeth [43]) Compared with SFIA

survey was coincident with the COVID pandemic and lockdowns. As all the data collection was collected remotely via an online survey, this did not inhibit data collection. Once the survey was closed, the results were downloaded in an Excel spreadsheet for data cleaning and analysis. Analysis was done within Excel, on hard copy with affinity diagramming, and in Atlas-ti.

Participant Practitioners and Experts

Participants were industry practitioners, with different levels of expertise and years of experience in software testing, ranging from under 2 years to over 20 years (Figure 1). Anyone mentioned as an expert will be evidently working at SFIA level 5 or above. A participant referred to as a practitioner will be working in industry and may be working at any of the SFIA levels.

Participant Recruitment

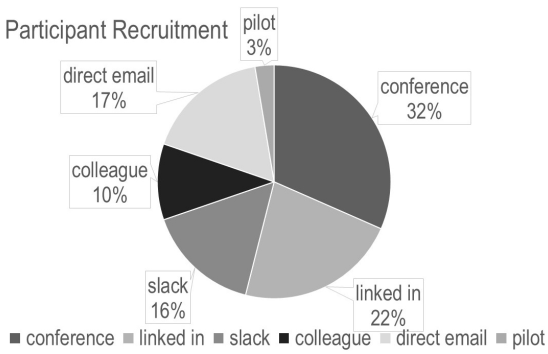

Fig. 2. Recruitment

Participants in this research opted in to answer an anonymous, open-invitation survey. The opt-in recruitment means participants were not chosen by the researcher but self-recruited, as information about the survey was shared. There was an international sample, and the online requests for participation allowed a potential global reach. Participation came both from those who had testing as the major or total focus of their role, and from others where testing was part of their role.

Figure 2 shows the distribution of participants across the different invitations: 32% found out about the survey at a conference, 22% from Linked-in posts, 16% from Slack communities, and the remainder by direct contact or via a participant's colleague. Figure 3 shows the dropout rate through the survey, at each stage. There was a low drop out,

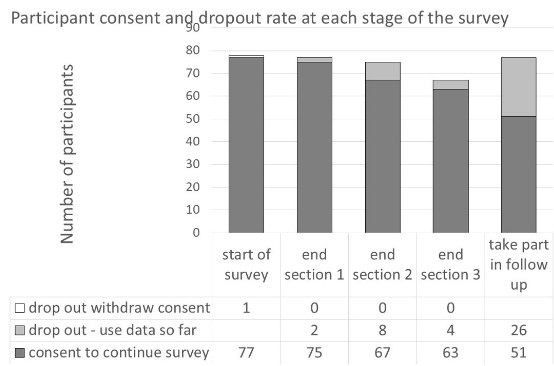

Fig. 3. Drop Outs

with only one participant asking for their data to be removed from the survey.

Data Preparation and Analysis

The data was cleaned, removing participants who had withdrawn consent for their responses to be used, and anonymising the remaining participants.

To provide a validity check for participant recruitment, the demographic data from the sample was compared with several industry surveys. The industry reports used were the World Quality Report [13], media packs from two major software testing conferences, [18, 57], and two test tool vendor industry surveys, [38, 47]. There were similarities in demographics to the sample.

For the demographic and closed questions, quantitative methods, for example, frequency analysis, were used. Qualitative analysis was used for the open questions and text responses, by mining for quotes, coding, and looking for groups and themes. An interpretivist stance after [6], was used when interpreting qualitative data from open questions.

The open questions included the participants' hobbies, qualifications, training, job titles, roles and approaches. Table 2 summarises the coding and themes used. These were based on the literature and grey literature review, and also emerged during the analysis process. Coding arose from how the participants described themselves, and their backgrounds, rather than imposing a coding scheme in advance. Several rounds of coding took place, as patterns emerged revisiting the literature and grey literature to see if the patterns were common elsewhere, adding codes as a result, and recoding, looking for additional patterns, codes and themes.

For example, when looking at the responses about training, it was noticed that courses might be referred to by their title or syllabus, by the name of the person who presented

Question topic	How categorised and coded
Hobbies	Coded to categorise by topic, level of activity, location, sociability, and recoded against codes and themes used in [39], including Artistic; Social; Stereotypical IT worker (low social, low arts).
Academic qualification	Coded to 5 level groups - (Doctorate, Masters, Bachelor, Pre-bachelor, not given) and Subject groups (Arts, Social Science, Science, IT, Other).
Training in testing	Code groups emerged from data as: 'named courses' e.g. ISTQB, BBST, 'named experts' and 'self-taught'.
Role Description	Coded in 3 groups (1) 'responsibilities' e.g. automation design, risk assessment, (2) 'approaches to testing' e.g. exploratory, scripted, and (3) 'techniques' e.g. state transition, happy path.
Aspirations	Coded as Low e.g. no wish to change, Medium e.g. looking to upgrade skills, run team or High e.g. CTO, international speaker
All open questions	Communication style e.g. 'terse', 'conversational', 'expansive', 'systematic'.

Table 2. Survey Topics: Qualitative Analysis - Open Questions (after [21])

the course, or by content, potentially allowing an insight into what different people valued in the training. Hobbies were initially coded based on observed patterns (indoor versus outdoor, team versus solitary) and then calibrated in a recoding based on the literature, which was searched to understand if any research had found patterns in IT workers' hobbies or interests, in particular [39, 59], using codes such as 'Artistic', 'Multi-Interested' and so on. This provided a rationale for the classification of hobbies, and an opportunity to re-code the data and validate the codes. For the domains and sectors, codes were taken from the divisions in reports such as [13, 47]. For academic qualifications, answers were coded using terms such as 'Masters', 'Arts', 'Science', and so on. Similarly, for training in testing, we adopted generic codes such as 'named courses' to code formal testing-related training such as ISTQB and BBST, and 'named experts', when participants mentioned experts they follow. Role descriptions were coded in terms of responsibilities (e.g., 'automation design', 'risk assessment'), people's writing styles (e.g. 'terse' versus 'expansive') and their approaches to testing and techniques (e.g., 'exploratory', 'state transition', 'happy path').

Answers to open questions in the survey provided rich stories of testers' work and home lives, in the participants' own words. Extraction of interesting quotes from these responses led to generation of codes which built into themes such as 'enthusiasm', 'aspirations', and 'meeting challenges.' Categorising the responses from both open and closed questions into groups such as 'type of training' and 'academic qualification' allowed further frequency counts.

Initial analysis was performed by the main researcher, with data and emerging findings cross-validated by the other authors, as well as at peer reviews at industry conferences, and examination of industry reports, standards and literature.

Research Validity and Ethics

This study has taken place at least partially within the field of Software Engineering, and has included collecting data from people working within the software industry. Both validity and ethical matters affect the outcomes of this research. In this section the threats to validity and ethics are considered, and the mitigations used are described.

Ethical Dilemmas in ESSE

A discussion of ethical concerns in empirical studies of Software Engineering (ESSE) by [54] raises issues raised by empirical research in Software Engineering, including use of surveys, interviews and case studies, all methods used in this study.

> "Our discussion is most relevant to research projects that employ human subjects or that involve the collection of information that can lead to the identification of individuals. Such identifiable information can be collected through the observation of humans or through the examination of artifacts (i.e., source code or documents). Note that this covers much of empirical Software Engineering research . . . " [54]

The ethical dilemmas are "four core research ethics principles" [54] which are applicable to this research and provide examples that point to specific ethical issues raised by the use of empirical methods in ESSE research:

- **informed consent**: all participants were given information about the study they participated in and asked for informed consent, as well as being given the option to drop out of the study at any time;
- **scientific value**: data collection and analysis focused on the research question, and addressing the knowledge gap identified. Where possible data sets and instruments have been shared on OSF to allow reproducing of the study methods;
- **beneficence**: focus on helping to improve industry practices for individuals and organizations; and
- **confidentiality**: Anonymising was done by coding each participant to a number for the study and participant, for example [SP10]. Some participants' records still contained opportunities to identify them and their companies. Therefore, it was decided not to share the dataset in replication packs, but instead to share the survey instrument used.

This research has been carried out under the University of Malta Ethical Guidelines and Procedures and approved by the Ethics Board[2].

Threat to Validity During Data Collection and Mitigation

Threat: The sample for data collection could be too narrow a sample to reflect the general population of testers.

Mitigation: Taken into account when recruiting for the survey, so that the sample participants were not just a convenience sample of 'people known to the researcher.' The survey was open-invitation, advertised at a range of conferences and events, as well as in on-line discussion groups. A check was made on the actual participant sample, by

[2] See `https://www.um.edu.mt/research/ethics/`

comparing demographics against demographics for two industry conferences, and three industry reports [57, 18, 47, 38, 13] to see that there was an approximate match.

Threat: This research has the characteristics of being both 'of me' because the researcher is a tester, and 'plus 1' because other researchers have worked on related areas.

Mitigation: Constantly being mindful of potential bias both in sampling and interviewing. Reflecting on each stage to understand how activities had been run. At each stage, looking for improvements in how to ask questions and how to interpret answers. Being a 'plus 1' study was helpful because similar studies could be used to inform question design and data analysis.

Threat: Sample size may not be representative. The Survey had a total of 78 participants. Not all the participants answered all the questions.

Mitigation: Qualitative methods were applied to analyse answers to open questions, using themes from literature review, or emerged from previous stages in this work, and also as they emerged from the data. Re-analysis to saturation was used to stop the coding, in other words, recoding proceeded until the results were not changing, following [6]. The sample sizes were reasonable for qualitative work [27].

Data Availability Statement

Survey data is not shared, because of the difficulty of anonymising the responses sufficiently for personal and commercial confidentiality [6, 54]. The survey questions are available in the Appendix, and available in a replication pack, in OSF project (`https://osf.io/e6zjn/?view_only=13a99944d3534ed19dc5136ea48792b3`.

Results

This section describes the characteristics of the testers who participated in the survey. First, the demographic data about age, gender, and location of the participants are described. The participants' backgrounds, including their academic qualifications and previous roles, are then described, followed by their current roles, job titles and aspirations. Approaches, techniques and tools that participants mentioned using in their work are described, then testing revisited as role versus as an activity. Participants' hobbies and interests are examined, to help understand them more as people. Finally the results of analysing all the open questions to understand communication styles and preferences, leads to the notion that stereotypical views of people in testing is less useful than observing their actual characteristics.

Participant Profiles

Demographic information was collected in the survey and used to assess whether the participants provided a sample that reflected the testing community.

The participant gender split was 46% male, 53% female, and 1% non-binary. The participants' ages ranged from under 25 to over 60, with over 90% between 25 and 64, so within ordinary work age.

The split between male and female genders is roughly even for the age range 25 to 64. However, the group aged 65+ is predominantly female, while the under 25's are all male. This surely does not reflect either age group. As these two age ranges are a small proportion of the whole participant sample in the survey, while they will be mentioned in the discussions, they are probably not indicative of anything significant in gender and age for recruitment patterns. A further study could focus on demographics as a factor in diversity, but for this study, the main concern was having a gender mix of people in a wide band of standard working age.

Participants were mainly from Europe (61%), North America (14%) and Asia (11%). A smaller group were from South America (3%), and the remainder from Australasia. An attempt was made to recruit more participants from underrepresented areas, with limited success. Countries with the most participants were the UK, Netherlands, USA, India, Canada, Denmark, Malta, Portugal, Brazil and France. There were no participants from Africa, despite requests to networking groups. The work of [61] suggests that studies of testing practitioners looking at location may provide further insights into their motivations and attitudes to testing. This is an item for future development of this work.

The survey was open for responses from September 2021 to July 2022, and therefore[3] although 90% of the participants were in full-time work, over 80% were working from home, with just under 40% making some visits to their work office premises, and just over 15% making some visits to client sites.

The participant profiles were compared with the profiles in published commercial industry surveys taken in the same time period, including the World Quality Report [13] two tool vendor surveys [47, 38] and the media packs for two international conferences [18, 57] to compare the demographic and job title information. There were similarities in the demographics and job titles across those reports to this survey.

Participants' Background

Participants were asked about their backgrounds, including their education and academic qualifications, their professional training including training in testing, and their experience in different roles.

Academic Qualifications

Academic qualifications were grouped by highest academic level, into end of secondary (high) school, leaving tertiary education before graduation, achieving a bachelor's degree, achieving a master's degree, and working on or gaining a post-Masters qualification. Table 3 shows academic qualifications by gender and age. About 80% of the participants had a Bachelors degree or above. Just under 4% were working on or had a Doctoral qualification. Most of the participants had either a Bachelors or a Masters degree, with just under 48% of male participants having a Bachelors, and just under 48% of female participants having a Masters degree.

Table 4 shows the percentage participants with different degree subjects. Just over 40% of the participants did not specify their degree subject, around 18% because they had no degree. Arts graduates made up just under 10% of participants, with 14% having read social sciences, 25% sciences, and just under 17% IT-related degrees. Testers include, but are not exclusively, Software Engineering graduates.

Previous Roles and Years of Experience

Participants were asked for their years of experience in different roles in IT. These were closed questions, participants selecting year ranges against different work areas, followed by an open question to comment on their answer if they wished to.

[3] COVID pandemic and lockdowns

Academic Qualification (Total all)	Male	Female	Non-binary	under 25	25-34	35-44	45-54	55-64	65 or over	N=
High School (9.1%=7)	8.3%	10.0%	0.0%			11.5%	11.1%	18.2%		7
Pre-Bachelors (9.1%=7)	13.9%	5.0%	0.0%	50.0%		11.5%		27.3%		7
Bachelor's degree (37.7%=29)	47.2%	30.0%	0.0%	50.0%	42.9%	26.9%	44.4%	36.4%	60.0%	29
Masters Degree (39.0%=30)	27.8%	47.5%	100.0%		50.0%	46.2%	38.9%	9.1%		29
Post graduate (3.9%=3)	2.8%	5.0%	0.0%		7.1%	3.8%	5.6%			3
no response (1.3%=1)	0.0%	2.5%	0.0%					10.0%		1
Total of group	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	
N =	36	40	1	2	14	26	18	11	5	76

Table 3. Survey: Academic Qualifications Against Gender and Age

Subject Area	% (N) of Participants	Roles and Approaches
Arts	9.9% (7)	42% in technical roles, such as automation engineer, and 28% in management roles. Conversational response styles. Multi-functional roles, team problem solving, diagramming.
Social Science	14.1% (10)	40% in technical roles, 50% in management, 30% consultancy/coaching. Conversational response styles. Multi-functional roles, seeing the 'big picture' in a project.
Sciences	25.4% (8)	44% in technical roles, 50% in management, 27% in non-technical roles. Terse response style, often as ordered lists.
IT	16.9% (12)	58% technical role, 41% non-technical role, 50% had management responsibility. Response styles were to the point, not conversational, but full of technical detail.
Not stated or Pre-Bachelors	42.3% (30)	31% in technical roles, 47% in management, 31% in non technical roles and 1.5% consultants. With an expansive, conversational style, this group had plenty of stories to tell about communication, pairing, and project-wide/non-testing aspects to their roles. This group used frameworks and methodologies when describing work approaches.

Table 4. Academic Subjects, Roles and Styles of Responses (N=77)

As noted in the Background, years of experience contribute to but are not the only factor in assessing expertise. While asking about years of experience is not going to provide information about how well those roles were done, or about the level of expertise, it does give an understanding for these practitioners of the breadth of experiences in their careers, and exposure to concepts from other IT disciplines. Many of the participants had multiple experiences across many roles.

The first group of roles could be seen as central to this research: testing, test management and development. Participants had at least some experience in software testing. This ranged from under two years (5%) to over 20 years (20%) with the largest group being the 37% with 5-10 years experience in testing (see Table 5). Four participants did not mark specific software testing experience in the closed questions, but in the open questions were clearly talking about testing work, whether in test design, results checking, or specialist testing such as security or usability. They have been shown in the results tables as 'no experience', because how many years is not clear. Their answers to open questions have been assessed in the analysis on their own merit. Most of the participants also had experience in test management, with 75% having had a role involving test management, and 55% some experience of test programme management roles. Over a quarter of the participants had no development experience, and another quarter had more than 5 years development experience.

The next group of roles are areas that are 'shift left' in focus, that is, the team, including testers, become involved in reviewing artefacts and ideas before coding starts. The purpose of this is to centre activities around understanding what customers and users want, and reduce potential defects being introduced into the product. As well as requirements, the product designs that specify how it is to be built are reviewed. These roles and activities include the modern roles of UX design and product ownership (see Table 5) as well as the traditional requirements, architecture and design activities (see Table 6). Traditionally, software testers have used requirements analysis and design analysis, as key techniques in their craft [8, 7, 42], both in order to uncover errors via static testing, and to derive tests for dynamic testing. It is therefore not surprising that 65% of the participants have some experience of requirements analysis, and around 50% in system architecture, analysis and design. Indeed, entry into software testing as a specialist role was possible from a business analyst and systems analyst role, as well as from a developer role, in the 1980's and 1990's[4]. However, a much smaller proportion have over 5 years' experience in these areas. There is less experience among the participants in the more modern customer-facing roles of UX design and product ownership. Over 50% of the participants had no experience in these roles.

Release management, operations and support experience was mixed, with just under 60% of participants having support experience (see Table 6). This is not unexpected, both as the move from support to testing is a career path, and also because testers can spend time in support as part of their induction into a testing role, so they better understand customer and user challenges[5] Just over half of participants had release management experience, and just under half operations experience. This might reflect career role moves, and may also reflect the modern multi-faceted DevOps roles, with team members taking on a range of responsibilities across a project.

Participants had experience in other management and coaching roles. Around half had some experience in risk, project and programme management. Over 60% had people

[4] Researcher's industry experience

[5] Researcher's industry experience

(N=78)	Development percentage (N)	Software testing percentage (N)	Test Management percentage (N)	Test Programme Management percentage (N)	UX design percentage (N)	Product ownership percentage (N)
Less than 2 years	21.8% (17)	3.8% (3)	7.7% (6)	12.8% (10)	29.5% (23)	24.4% (19)
2 - 5 years	24.4% (19)	12.8% (10)	17.9% (14)	19.2% (15)	14.1% (11)	15.4% (12)
5 - 10 years	11.5% (9)	35.9% (28)	28.2% (22)	16.7% (13)	2.6% (2)	2.6% (2)
11-20 years	6.4% (5)	23.1% (18)	12.8% (10)	1.3% (1)	1.3% (1)	1.3% (1)
Over 20 years	6.4% (5)	19.2% (15)	9.0% (7)	5.1% (4)	0.0% (0)	1.3% (1)
no experience	29.5% (23)	5.1% (4)	24.4% (19)	44.9% (35)	52.6% (41)	55.1% (43)
any experience	70.5% (55)	94.9% (74)	75.6% (59)	55.1% (43)	47.4% (37)	44.9% (35)
more than 5 years	24.4% (19)	78.2% (61)	50.0% (39)	23.1% (18)	3.8% (3)	5.1% (4)

Table 5. Experience: Development, Testing, Test Management, UX and Product Ownership (N=78)

(N=78)	Requirements analysis percentage (N)	System architecture percentage (N)	System analysis percentage (N)	System design percentage (N)	Release management percentage (N)	Operations percentage (N)	Support
Less than 2 years	7.7% (6)	25.6% (20)	15.4% (12)	24.4% (19)	26.9% (21)	26.9% (21)	32.1% (25)
2 - 5 years	20.5% (16)	12.8% (10)	23.1% (18)	11.5% (9)	16.7% (13)	12.8% (10)	17.9% (14)
5 - 10 years	24.4% (19)	6.4% (5)	10.3% (8)	10.3% (8)	7.7% (6)	2.6% (2)	5.1% (4)
11-20 years	9.0% (7)	2.6% (2)	2.6% (2)	2.6% (2)	0.0% (0)	0.0% (0)	1.3% (1)
Over 20 years	3.8% (3)	1.3% (1)	3.8% (3)	2.6% (2)	2.6% (2)	2.6% (2)	1.3% (1)
no experience	34.6% (27)	51.3% (40)	44.9% (35)	48.7% (38)	46.2% (36)	55.1% (43)	42.3% (33)
any experience	65.4% (51)	48.7% (38)	55.1% (43)	51.3% (40)	53.8% (42)	44.9% (35)	57.7% (45)
more than 5 years	37.2% (29)	10.3% (8)	16.7% (13)	15.4% (12)	10.3% (8)	5.1% (4)	7.7% (6)

Table 6. Experience: Requirements, Design, Release Management, Support, Operations (N=78)

management, agile team leadership or coaching experience. Just under one third mentioned other roles (see Table 7).

Backgrounds Outside IT

While many participants only described their roles within IT and software, there were also other backgrounds that people brought into their testing roles; what one participant referred to as 'roles outside tech'. Some of these backgrounds were very obviously related to IT, including testing: maths, applied science, engineering, electronics, and electronic engineering. Other skills and knowledge potentially very useful to enhance testing included technical writing, science communication, languages, linguistics and international relations. These all have the potential to improve communication in and between teams; communication was raised by participants in the survey, and in industry publications such as [47] as critical for software testers in roles that span across the organisation. Knowledge of the domain for the software under test, as well as participants' understanding of what is critical for organisations relying on software, is perhaps enhanced by having a background in the domain, HR, recruitment, business administration, marketing, or economics. Philosophy, science, urban planning, medicine and maths all provided participants with a training in critical thinking, problem solving, analysis of complex problems that are likely to enhance a tester's overall skill set. With the strains and stress of projects, the less obviously useful skills brought in by a handful of participants, the psychologist, therapist, life coach, and yoga teacher, all have a place. The craft skills, patience, attention to detail, and focus on measurement contributed by a background in theatre, music, boat building, landscape gardening, or carpentry might all contribute to the attitude and mindset for patient testing.

Training Towards The Role

Over 70% of the respondents had attended formal training courses for testing. Other training options commonly mentioned included support through the testing community (5% via conferences, meet-ups, online community), learning on the job (25%), and learning from other fields. Formal training courses included the ISTQB courses, which provide a certification route from Foundation through Advanced to Expert level courses with exams. Participants' responses mentioned *'Effective Methods of System Testing, … Rapid Software Testing; Getting a Grip on Exploratory Testing; Black Box Software Testing (BBST) Foundations'* [SP1], while another listed *'ISTQB Certified Tester Foundation Level, BCS Requirements Engineering'* [SP58]. Named experts were also mentioned by 5% of participants as providing courses. Some 8% of the participants had received no training at all.

Participants' Work Profile

Data was collected about participants' work profiles in both open and closed questions, about their role, the approaches they used in the role, and whether testing was a part of their role, or the role was testing. The participants wrote about whether their roles and responsibilities matched their job title, and their aspirations. They were asked to describe their approach to testing, with the techniques and tools they use. They were asked to describe what else they did apart from testing.

In addition to years of experience, the open questions on approaches to testing gave insight into changes with experience. For instance, [46] shows experts use visualisation; participants with longer experience mentioned visualisation, whereas less experienced participants did not: 'I diagram or mind map just about everything' [SP1], and 'flow diagrams, context diagrams, mind maps, story mapping' [SP68].

(N=78)	RM %age (N)	PPM %age (N)	Agile TL %age (N)	Coaching %age (N)	People lead %age (N)	Other %age (N)
Less than 2 years	21.8% (17)	20.5% (16)	17.9% (14)	24.4% (19)	20.5% (16)	15.4% (12)
2 - 5 years	11.5% (9)	14.1% (11)	24.4% (19)	11.5% (9)	12.8% (10)	9.0% (7)
5 - 10 years	7.7% (6)	14.1% (11)	11.5% (9)	20.5% (16)	20.5% (16)	3.8% (3)
11-20 years	6.4% (5)	3.8% (3)	3.8% (3)	2.6% (2)	6.4% (5)	0.0% (0)
Over 20 years	1.3% (1)	1.3% (1)	1.3% (1)	6.4% (5)	6.4% (5)	2.6% (2)
no experience	51.3% (40)	46.2% (36)	41.0% (32)	34.6% (27)	33.3% (26)	67.9% (53)
any experience	48.7% (38)	53.8% (42)	59.0% (46)	65.4% (51)	66.7% (52)	30.8% (24)
more than 5 years	15.4% (12)	19.2% (15)	16.7% (13)	29.5% (23)	33.3% (26)	6.4% (5)

Management Leadership and Coaching (N=78):
(RM = Risk Management PPM = Project and Programme Management TL = Team Leadership)

Table 7. Experience: Management Leadership and Coaching (N=78)

	Scale											
Not at all	0	1	2	3	4	5	6	7	8	9	10	**Completely**
	0	0	1	4	0	3	3	12	21	9	17	
	Number of Responses											

Table 8. Survey: Does Job Title Match to Role?

Role, Job Title and Aspirations

Participants were asked to describe their current role, provide their job title, score their job title for fit to actual activities, and comment on their job title. Across the participants, five held directorships, and another three head of department roles. Another 45 were working in manager, lead, principal, senior or specialist roles. The job title with the roles description showed they were clearly working at SFIA Level 5 or above.

Table 9 shows the number of years more senior participants had in five roles: tester, test manager (TM), test programme manager (TPM), development, and UX. The upper part of the table shows the numbers for all 53 participants considered at the end of the data analysis to be in roles at SFIA level 5 or above. The lower part of the table shows the same analysis just for the 8 participants in director or head of department roles. As might be expected, all these senior practitioners had considerable experience in testing and test management activities. Interestingly, they all also had at least some experience both in development and in UX activities. The director and head of department participants showed a slightly different pattern; a larger proportion of people with no development, and no UX experience. This is not a large enough group (N=8) to make any definitive claims, but it could be indicative that senior leadership in testing might lack knowledge of people-focused and design thinking methods, which could be useful as they set the direction for recruitment, team building, staff development and tooling. Investigating this further would be useful future work.

When describing their current roles, participants talked about level of seniority, role focus and job title. Testing and engineering were the most frequently mentioned terms, test engineers, quality engineers and automation engineers all being mentioned, either as the participant's job title, or the job title of people they work with. One participant commented that *'the word engineer makes me wonder...'* [SP3], while another wrote *'Every test specialist in the company is called a QA Engineer, but with different actual focus . . . '* [SP14]. It was not clear from participants' descriptions of their qualifications, professional training or role descriptions whether 'engineer' was used in the sense of chartered, incorporated or technician level as one would find on, for instance, the Institute of Mechanical Engineers website membership page[6]. The BCS website has a list of the necessary skills for a software engineer, rather than specifically for a test engineer or a quality engineer, which include communication, planning, and coding, but does not have the same distinction of chartered versus technician roles.

Additionally, roles are changing; testing is becoming (again) an activity done by people across the team, rather than a role or separate team. In the last year in communities such as Ministry of Testing[7], there have been vigorous debates about the role of a tester and its change in some organisations to Quality Engineer, Test Engineer, or Quality Coach. These are roles that advise and assist, but are not responsible for carrying out the testing themselves. This is particularly true in teams running agile and DevOps projects, with testing carried out by all the team, sometimes in pair or ensemble mode. Developers, product owners and specialist testers work together. The implication is an even wider range of potential personas. About 10% of participants wrote about coaching as part or all of their role, plus approximately 30% engineers (8 quality engineers, 11 test engineers, plus some DevOps and automation engineers). Other people looked like they did the same job, but with a different title, for example automation tester rather than automation engineer. Job titles, but not necessarily roles and activities, are changing.

Participants often said their roles were complex with multiple responsibilities, in the words of [SP16] they *'wear a lot of hats'* while others such as [SP37] combined multiple roles; testing, automation, agile team leadership, coaching, communicating with the business, and coordinating workplace guilds.

The aspirations of the survey participants for career development were wide-ranging. Some people stressed that they were happy in their current role with no aspirations for career development; typically because they were recently appointed to their current position, and therefore it was too soon for the next

[6] See `https://www.imeche.org/membership-registration/become-a-member` (Accessed 26 February 2025)

[7] See `https://www.ministryoftesting.com/news/exploring-quality-engineering-the-testing-planet-ep-08` (Accessed 26 February 2025)

SFIA 5 or above (N=53)	**Testing**	**Test Management**	**Test Programme Management**	**Development**	**UX**
none	0	0	0	0	0
Under 2 years	0	1	5	10	13
2-4 years	4	11	11	13	8
5-10 years	19	16	10	6	1
11-20 years	13	7	1	4	1
Over 20 years	12	7	4	4	1
Director / HoD (N=8)	**Testing**	**Test Management**	**Test Programme Management**	**Development**	**UX**
none	0	0	2	1	6
Under 2 years	0	0	0	3	1
2-4 years	0	0	0	3	2
5-10 years	4	5	4	0	0
11-20 years	2	1	1	1	0
Over 20 years	2	2	1	0	0

Table 9. Survey: Senior Roles' Experience in Different Areas

stage. Those who did express aspirations were very varied in their ambitions; around two-thirds wanted to grow within their immediate roles, learn new skills, and lead their teams. Other noted the need for continuous professional development simply to stay up to date with skills: *'Keep learning and adapting to the changing world in testing and maintain a position in Quality Assurance'* [SP69]. Some were already at C-level roles, others aspired to an eventual C-level role, or equivalent industry influence; around 12% expressed ambitions at that level. The SFIA model [25] only takes testing roles up to level 6, although related roles such as quality management do reach level 7, so the mention of CEO (Chief Executive Officer), CTO (Chief Technology Officer) and CQO (Chief Quality Officer) roles perhaps indicates a direction for updating career progression models.

Many participants expressed a desire to help others - locally in their teams, in their organisations, and with their customers. Several also talked about aspirations to international conference speaking, or sought to influence across the industry; responses to these questions showed enthusiasm, aspirations, and a lack of boredom about testing.

Approaches, Techniques and Tools

Subject	**Topics**	**Percentage**	**Examples**
Testing	229	73%	BVA, Exploratory, Test(ing), Automat(ing), Analyz(ing)
Communication	61	19%	Listen, Talk, Ask, Answer, Support
People	14	4.5%	Inspire, Mentor, Coach, Teams
Management	11	3.5%	Schedule, Plan, Manage

Table 10. Survey: Testing Approach and Techniques

The participants' responses about the approaches, techniques and tools that they use were not unexpected; a focus on activities, approaches and techniques that clearly support designing and executing tests, and analysing their results. This included use of test design techniques such as equivalence partitioning, boundary value analysis and pairwise. Participants also mentioned exploratory and context-driven approaches to testing, as well as mentioning scripting for regression testing and feeding automation tools. Heuristics were a choice for both designing tests and for understanding the results of tests. Communication and 'glue work' to ensure different stakeholders and team members understood each other also featured, especially for the more senior roles. As participants moved through the SFIA levels, their roles changed from doing to influencing.

While over 70% of the topics raised were directly about testing, 23% were directly related to communication and people. Verbs such as listen, ask, answer, talk were frequently used, along with coach, mentor, support, encourage, and inspire (see Table 10).

Over half of the participants did not answer the question about what tools they used. Of those who did answer, test automation tools were mentioned, but were not exclusively the focus for comments on tools. Participants also supported their testing by using tools to help with planning, test design, test and data creation, analysing and reporting on test results. Specialist tools for aiding accessibility, performance and security testing were in use. Tools to support thinking, problem solving and communication of ideas were also mentioned, for mind-mapping, brainstorming, identifying and discussing risks, and communicating proposed testing. One participant mentioned introspection as an important activity, supported by tools that map app behaviour to business outcomes. Testers also made use of the tool set natural to a development team, such as GitHub, Azure DevOps, SQL. For these testers, tools also included white boards, paper and pencil, and their own brain. Nemeth's modelling of the roles clearly matched, with participants describing knowledge-based behaviours that were problem solving, pattern recognising, and decision making, as well as rule-based and monitoring behaviours such as monitoring tool-supported testing (see Table 1).

Hobbies and Interests

Participants showed a wide range of interests, both across the group, and as individuals. Table 11 lists the hobby groups from most mentioned (outdoor, sporting and fitness) to the

Hobby Group	Survey %
Sport, exercise, fitness, outdoors	60%
Making, performing, writing, playing	45%
Reading, studying, learning	41%
Watching/listening theatre/music	28%
Family and friends	26%
Watching film, cinema and TV	17%
Games and puzzles	17%
Food and drink	15%
Travel, other cultures	13%
Volunteering and community	8%

Table 11. Hobby Groups: Percentage of Respondents (after [21]

least mentioned (community and volunteering). The outdoor pursuits included running, gardening and bird watching. The second most mentioned group was the creative group: makers, performers, writers and musicians.

Hobbies and interests were coded into groups to indicate participants' level of active involvement, proportion of the activities that were indoor to outdoor, art versus STEM and individual versus team activities. From survey responses: Listening to music is a relatively passive, arts activity that is likely to be indoors and individual. Singing in a choir is an active, arts activity that is likely to be indoors and is a team activity. Composing music is a very active, indoors, arts, individual activity. Visiting a music festival with friends or family is a relatively active, outdoor, team, arts activity. Table 12 displays the data for all survey responses. In the initial coding, each response was examined to differentiate between hobbies that are (a) more passive and more active, (b) more indoor and more outdoor, (c) more arts or more STEM, and (d) more individual and more team-based (see Table 12). For each mention, at this stage, a subjective score between 1 and 10 is given based on the content of their response (e.g. a score of 10 for the Indoor/Outdoor range is given if all the hobbies mentioned can only be carried out outdoors). Participants could be in more than one group, for example, with active artistic and passive STEM hobbies. An example of a passive indoor hobby is one participant who simply put *'watch netflix'* [sic] [SP11], while an active, multi-faceted response was *'i love music and non-mainstream cinema; i travel a lot and go to music festival; i'm passionate about people and love to spend time with my friends; i'm very active, always doing stuff and planning new adventures'* [sic] [SP3]. An example of a more STEM-related hobby was *'Photography, forensic anthropology and medical books, word games, escape rooms, travel'* [SP15] while *'Painting, kickboxing, feminism, reading, dancing, volunteering, making costumes, creating in general'* [SP37] is someone active in arts and community. The group scored more highly for active, indoor, and artistic hobbies and had a balance between individual and team hobbies.

In further calibration, analysis of the survey responses was then made using the themes and codes from [39], which compares the interests with those in O*NET, a recruitment database. This provided additional insights. Table 13 shows the results from the survey, compared with [39]'s results for 500 IT workers and 500 aspiring IT workers, across different disciplines. The O*NET stereotype for an IT worker is for a person to score low on social and arts interests and to have a low diversity of interests. The results in [39] show that 30% of the actual and 36% of aspirant IT workers meet the recruitment stereotype, so around three quarters of the people in their sample did not meet that stereotype. Within the sample of testers participating in this survey, only 6% met the recruitment stereotype in their interests.

Communication Styles and Preferences

Table 4 shows the communication styles found in the responses to open questions, set against the academic subject participant studied. Arts graduates had a conversational style, often referred to teamwork, and used diagramming in their work. Social science graduates also had a conversational style, and talked about having multi-functional responsibilities, organisational issues, and seeing the "big picture' in a project. The science graduate often used ordered lists to respond, in terse but understandable style. IT graduates often had short responses, to the point, were not conversational, and supplied a lot of technical details, assuming understanding of those technical details. Participants who did not have, or did not state they had, tertiary qualifications often had an expansive, conversational style, were interesting storytellers, with anecdotes about teams and projects. This group referred to frameworks and methodologies when describing work approaches.

Discussion and Conclusions

This work was driven by the question: '**Who is Testing?**' and a wide diversity was found of people and backgrounds in industry. The emerging picture is complex: simple persona sets are not immediately obvious; instead there is a picture of variable characteristics, affecting how a person approaches testing, their motivations and the support they need. Even more important, people performing testing cannot be stereotyped; there is evidently a wide variation and diversity of people in roles where testing and related activities are significant. In the discussion, first the question of who is testing is considered, and then the question of why that matters.

Who is Testing

Diversity of Backgrounds

The diversity of backgrounds and experiences that testers bring to teams is evidenced in the data analysed. Although a number of respondents had entered testing by what might be seen as a conventional route of a Software Engineering degree, testers also entered their roles with other degrees, both science and arts, or no degree at all (see Table 4). Unexpected backgrounds included philosophy, medicine, theatre, music, physics, HR, urban planning, pharmaceuticals and boat building: *' . . . realised I didn't want to be a boat builder after all so . . . took up testing . . . that's where I've been ever since . . . '* [SP42].

Diversity of Hobbies

This diversity continued to be evident when examining testers' hobbies in the survey, with participants listing a wide range of types of interests, with many actively engaged in making and doing. For example, many respondents listen to music (29% mentioned it specifically), and some make music (12% singing or playing), and even in one case compose. Creative visual design and making were frequently mentioned (27%), with outdoor and exercise-related hobbies the most frequently mentioned: gardening to horse riding, running and kickboxing (61%).

Min	1	2	3	4	5	6	7	8	9	10	Max
Passive	1	2	2	4	13	19	15	13	3	0	**Active**
Indoor	2	10	6	10	7	22	10	3	0	0	**Outdoor**
Art	2	4	20	27	13	2	1	2	0	0	**STEM**
Individual	4	14	13	7	15	6	10	2	0	0	**Team**

Darker color indicates greater number of participants scoring on scale point.
Table 12. Range of Hobbies Mentioned - Heatmap (after [21])

	McChesney	McChesney	Our survey
Sample Size:	500	500	71
CS/IT role:	Employed	Aspiring to	SWT
Artistic	20% (72%*	41% (60%*)	91% (53%*)
Practical	not given	not given	66%(51%*)
Analytic	not given	not given	38% (48%*)
Social	not given	not given	28% (32%*)
Multi-interested	19% (27%*)	not given	74% (54%*)
Few interests	31% (31%*)	23% (44%*)	23% (35%*)
Stereotypical IT	30% (30%*)	36% (36%*)	6% (50%*)

* Of which are female/non-binary (after [39]).
Table 13. Tester Hobby Types Not Stereotypical (after [39] and [21])

Comparing [39]'s figures with the hobbies information collected from software testers in the survey (see Table 13), and re-coded against [39]'s themes, testers scored highly on practical (66%) and arts (91%), with people generally having a range of interests (74%). This was not connected to degree subject; the pattern was similar across subjects. Software testers appear to be more artistic and to have a wider range of interests than the group of IT workers [39] studied, who themselves do not conform to the O*NET stereotype for IT workers. Only 6% of the survey participants conformed to the O*NET stereotype for IT professionals. Additionally, the range of hobbies broke outside the 'boring' stereotypes in [59]. There did not appear to be gender bias; roughly half of the artistic participants in our work identified as male. However, [39] do not report on all the ranges of activities covered in our work, as shown in Table 13. The table shows the comparison where it could be made.

Hobbies such as singing in a choir or playing in a sports team are indicative of an orientation to team-based work activities such as ensemble or pair testing. The O*NET stereotype of an IT worker's interests was for a low score on social activities, indicating a solitary nature. The range of communication styles and hobbies (see Tables 4, 12 and 13) show a group of people together able to meet the challenges of teamwork, remote working and customer focus within modern organisations.

The pattern in the data is one of divergence from the perceived stereotypical IT worker described in both O*NET and [39]'s work. The participant testers reported many social engagements, with family, friends, fellow hobbyists and in the testing community. As only 6% of participants conformed to the O*NET stereotype, compared with 30% of [39]'s sample, it might be expected that most of the participants' work behaviours and communication style to be non-stereotypical. There was emphasis on pairing, teamwork, diagramming, communication and testing as a social and creative activity, as well as a politically and cognitively challenging one.

Learning and Aspirations

The data shows that testers have the ambition to seek improvements in their own work, and also in the software and systems they help deliver.

Training impacts testers [41]; in this study, over 50% of those who discussed training appreciated a mix of ways of learning, for example: *'Maaret on Exploratory Testing Foundations course . . . one of the best trainings . . . loved Fiona Charles's masterclass on Test Strategy. . . . hours and hours of self-learning . . . attending conferences . . . practicing.'* [SP2]. This emphasis on learning may be of significance considering the call from C-level executives in [13] for greater skill levels among testers as the **main factor in improving productivity**. Learning also reflects the testers' stated aspirations reaching to an ambition for C-level roles, and extends to wanting industry-wide influence, stepping up to the challenges laid down in the World Quality Reports [13, 14] and State of Testing Reports [47, 48] in recent years. This study reinforces the findings in [24], in their call for people within and outside computer science to have more and varied education in testing.

Communication Skills

In this group of participants, there are people clearly working at expert level, SFIA level 5 to 7, and as knowledge-based in their skill sets and activities (see Table 1). Among the practitioners and the experts, aspirations to career progression were expressed in several directions: technical, managerial, industry-wide, and C-level within organisations. These participants were not all IT or Software Engineering graduates. People from different academic backgrounds appeared to provide a range of communication styles and this is potentially of great benefit to support improved communication between IT and its stakeholders, as well as within software projects. Those people with good communication skills may be better placed to successfully enter those higher-level roles than those with good technical skills but lower communication skills. Working with C-level executives in a way that helps them understand how to meet their challenges requires the ability to communicate technical, social, ethical, societal, customer and business concerns in a way that is understandable, digestible and actionable. Given these factors, it could be beneficial to explore how models for career progression, whether via professional membership bodies or skills frameworks, can support the aspirations of those working in testing, from many backgrounds, to progress.

Are Testers Different?

The data suggests that testers possess a diverse range of interests and display a distinct profile. In contrast to participants in [39], where 30% met the O*NET stereotype, only 6% of the participant testers were stereotypical: this survey found testers tend to have multiple interests, engage in artistic and outgoing hobbies, deviating significantly from the stereotypical classification found in O*NET and in [39].

Indications from this survey (see Table 4) and literature [28] are that variation of background may bring different communication styles and approaches to work. This informs organisational culture, teamwork and cooperation, encouraging better information exchange. Whereas software developers tend to originate from technical training, testers' education is rooted in a variety of different fields. The barriers to entry for testing seem to be lower than for other roles. For example, whereas [16] reports on successful efforts training veterans to take up roles in software testing, a similar initiative for doctors in the UK to learn coding in [31] seems less successful.

The ability of the profession to attract people with such diverse backgrounds stems from its multi-faceted and distinct nature. The survey data shows a spectrum of roles and responsibilities within the field, while [29] found that 91% of its participants were attracted to the job due to its variety. A study amongst 5,971 developers by [40] showed that most of a developer's day is spent coding (17%), bug fixing (14%), attending meetings (15%) and answering emails (10%). On the contrary, [56] observes that testers spend half their day communicating and learning how to mitigate the fact that they have to bring bad news to other project members. They also invest considerable effort in facilitating communication between different types of stakeholders. This is in addition to designing tests, implementing test frameworks, executing tests and managing the test process. This is echoed in our participants' responses, for example, *'My roles have required me to communicate a lot within my team, but have also involved speaking with the business, getting their feedback.'* [SP37].

Therefore, whilst software developers are likely to have similarly diverse interests to software testers, the latter cohort is more diverse in terms of its members' origins and the range of tasks that they need to complete on a day-to-day basis.

Why this Matters

To reflect on why these results matter, three areas are significant. It matters for society, for recruitment and on-boarding of testers, and for work practices and tools used in testing. Drawing on the concepts discussed in the Background Section and the Results, there are indications that stereotyping might be problematic, and this highlights the importance of diversity. Testing is challenging, requires a diverse group of testers and because of that diversity, the design of work practices and tools could be supported by HCI approaches such as personas.

Society

Both [30] and [53] represent HCI and AI communities which are increasingly calling for consideration of diversity. Supporting the testing community as a diverse and heterogeneous group helps encourage the notion of diversity in the IT industry. The warning from [30] that *'applying categories can bear the risk of labelling users as 'the other''* means care is needed to ensure that a diverse group of testers are neither ignored, nor encouraged to become a homogeneous and more stereotypical group of IT workers.

Product designers need to understand and support a heterogeneous and diverse customer population, and that is supported by a heterogeneous tester population representing them [30, 39]. The results of this survey show that a diversity of background exists in software testing, with a much closer match to the general population than studies of other IT practitioners. However, it is possible to increase the diversity further, and fully reflect the general population. Additionally, but beyond the specific scope of this work, it is important to distinguish stereotyping from acknowledging diversity, whether looking at roles, cultures, gender, or other factors.

There is then a call to action to ensure that people within IT, especially those who are testing software, represent not a narrow stereotype of the IT person, but as wide a diversity of people as possible. This could be impacted by recruitment and training of testers.

Recruitment, On-boarding and Training

The recruitment of testers is potentially problematic, because of stereotyping of background, communication styles, and other traits, as well as perceptions of tedium [39, 59, 61, 3, 22, 23] which match neither the reality nor the requirements of modern software testers. Testers come from diverse backgrounds, and this rich mix of people, together with the challenging and multifaceted roles they perform, could incentivise a wider population to take up a career in testing, to the benefit of society. Both academic and industry sources [13, 47, 53] state that a diverse and multi-skilled group is essential for testing. Encouraging a broad range of candidates to enter testing, at SFIA levels 1 to 5, and supporting progression to quality engineering at SFIA levels 5 to 7 can fulfil the requirements of organisations, and benefits society. Providing an understanding that the group already in the industry is diverse may engage and encourage even more diverse groups to apply, and to be recruited [39].

The skills, training and development required by testers throughout their careers follow from their recruitment to staff retention and development. Studies by [15, 62] have identified personality traits that best fit the requirements of various roles and tasks in Software Engineering, including testing. This survey data indicates that such efforts may be misleading, and with the stereotyping from recruitment and career advice indicated in O*NET may lead to essential skills and aptitudes being lost to IT projects.

The use of language abilities as well as or instead of numeracy as a predictor of programming ability, is remarked on by [28]: 'This is interesting because we as a field typically stress the fact that mathematical skills are important' and the finding that **language abilities were actually the better predictor of success at programming**. A wider-ranging recruitment policy both to colleges and by employers could benefit both the industry and society, by increasing both programming, testing and communication skills.

C-level executives have aspirations for the test community as ‘custodians of . . . quality at speed . . . focusing on value and on avoiding the possibility of defects’ [13]; testers’ diversity of thought and productivity to support this could be fueled by methods and tools that support efficiency, effectiveness and preferred ways of working. Participants in this survey indicated both aspirations to increased technical excellence, promotion to management, improvements in customer experience, and for some C-level / SFIA level 7 engagement with their organisations and the industry. This diversity of aspiration is important in keeping a healthy mix of different career paths open.

Successful recruitment, on-boarding, training and retention of good testers includes both removing stereotyping and providing career progression paths. This is also a call to action, to make the experience of working as a tester as satisfying as it can be, by making both work practices and tools highly usable, to support people’s effectiveness, efficiency and satisfaction [33], with work practices and tools that support both the technical needs and the overall UX of people testing.

Test Work Practices and Tools

The maturity of UX practices described by [44] is partly judged by the understanding of the people for whom a tool is designed. With a heterogeneous target group, this can provide challenges to the designer. [63] discusses the use of a basic set of personas to cover widely diverse groups in what can, to outsiders, look like a homogeneous group. In their case, the group is older adults, and in this study, it is software testers. The development and use of a set of personas allows an initial understanding of the target group. For testers, this could include themes and groupings identified in this study, such as communication styles and teamwork preferences. While a simple group of tester personas cannot easily be identified, work is in progress to provide a framework of heuristics, idea-t, which includes themes and groupings identified in this study, such as communication styles and teamwork preferences (see the Conclusion).

Communication needs and styles among testers differ more widely than those of other IT workers (see Tables 11, 12, and 13). The mix of practical, analytic and artistic hobbies indicates potentially a wide range of ways of communication styles and problem-solving preferences (see Table 13). Both tool interface design and workflow design would be affected by broadening understanding of tester personas to include preferences found. For example, a team-oriented, social person will favour workflows and tools that support pair and group problem solving; not all test tools support this well as was shown by [20]. The findings of this survey reflect those of [46], showing work practices and tool interfaces with strong visual elements may suit experts; maybe especially those with a leaning toward visual arts. A spreadsheet-style approach may be more suitable for testers with an analytic preference, indicated by hobbies.

The characteristics of testers, their backgrounds, capabilities and needs inform the characteristics of the workflows and tools they require. Automation and AI won’t remove the need for testing skills. In fact, all the indicators are that testers with increased skills will be given greater and more wide-reaching responsibilities for overall quality [14, 48]. Industry testing experts recognise this: [38] remarks: *‘We all recognize that . . . quality professionals play an essential role in building market-leading products and services. . . . A diverse group of people will be needed to carry out those challenging activities’*. Their activities must be supported by workflows and a toolset that enables them to maximise their productivity and capability to become the champions of quality while meeting their own C-level, customer facing or technical aspirations.

The need to support for a variety of people with different characteristics is supported by this work, and also by industry reports; *‘It’s clear that QA will lead testing, but it’s also beneficial to work toward democratizing testing with an everyone involved approach to create a culture of quality.’* [38]. Research from [61], [39] and [59] indicates why the vital role of testing may not be popular. This paper may counter those perceptions positively.

Future work is required to engage with developing the building blocks of improved tools and workflows for testing, better recruitment and retention of good testers, and the chance to reap the benefits of those improvements at all levels.

> *“Testing is harder than developing. If you want to have good testing you need to put your best people in testing.”*
> Gerald Weinberg[8]

Contributions

The first contribution of this study is to provide a greater understanding of the software tester community as a heterogeneous group, from varied backgrounds, and with differing communication, learning and work preferences. This builds on and adds to the work of [24].

The second contribution is built from the work in [19, 20] which identified that a narrow view of usability by test tools designers, that perhaps could be mitigated by a set of tester personas available to tool designers; this studied demonstrated that it was too difficult to build a simple set of tester personas that encapsulate the risch variety in the community.

This led to the work to build our next contribution, operationalisation of these results in a framework of heuristics and guidelines, idea-t, intended to help tools designers better understand the testers and context for which they design their specific tools. This framework has been subject to both industry case study and expert reviews, which are the subject of a later publication. The GitHub repository with the current version of the idea-t framework is found at `https://github.com/hci-lab-um/heuristics-for-test-tool-design/blob/main/README.md` where it is in development.

Future work

This research team is continuing work on a framework of heuristics to help people design, build and acquire tools to support testing. It will support more informed decisions about the personas who are the target testers using the tool.

Further work on diversity in testers might include increasing understanding about existing levels of diversity, and also into what diversity is required to reflect the general population when testing software and systems. This could include a meta-study based on works referenced in this paper and other papers that examine specific tester populations in different industries and locations.

Additionally, understanding more about stereotyping of testing roles both within and outside the IT industry could impact recruitment and retention. As part of this, the responses from our participants indicate a range of work style and

[8] quoted by `https://testguild.com/tribute-gerald-weinberg/` (Accessed 26 February 2025)

communication preferences, which are areas for HR and management research.

Understanding what education, training, and domain work experience best prepare someone for a testing role is a question for academic and industry research into the development of suitable qualifications.

Continuing to examine who is testing will help inform a better understanding of test workflow, approach and tool design, which in turn supports more diversity and heterogeneity in the testing done by software development teams.

Conclusions

The activity of software testing is challenging, requiring continuous learning, problem-solving, critical thinking and flexibility. The role of testers is changing, with a widening of responsibility and influence.

Testers are a diverse, interesting, and lively group of people with aspirations to improve their organisations and themselves, to better support their customers. Using qualitative data collected from industry practitioners, this study has shown that testers are not stereotypical IT workers, but are heterogeneous. This is a benefit that meets technical, organisational, and societal challenges. Whether originally artists, engineers, or computer scientists, they do not conform to stereotype, and require support, workflows and tools that enhance their unique challenges, preferences and capabilities.

Testers are well placed to meet the expectations of C-level executives quoted by [13, 14] in terms of influence and argumentation for increased quality at all levels in their organisation. Diversity helps meet the ethical challenges of increasingly important technologies, such as AI, and supports [53]'s call for 'muddy boots pragmatists' who remove bias by providing a diversity of views. The range of skills required indicates teams made of a heterogeneous group of individuals. Encouraging entry to the industry from as wide a group as possible is healthy for the testing industry, the wider IT industry, and society, enabling teams with diverse aptitudes, skills, and styles to work together. Breaking the stereotype strengthens the industry's ability to deliver what is needed by society and by customers.

These conclusions do not claim to establish a pattern in the population. Quite the contrary, the results indicate that the testing community is a diverse one which cannot be pigeonholed into one bucket called 'testers'. With regards to internal validity, the methodology is independently replicable and the researchers believe that the results discussed represent the characteristics of this specific cohort of participants. This claim follows from the research protocol, which included systematic efforts of data cleaning, coding and most importantly, regular validation with the community.

Several areas of interest for future research are identified. Understanding the range of test personas, their work styles and communication preferences, informs IT methodologies and tool design. Further understanding of stereotyping would help organisations reflect the general population when recruiting testers, positively impacting recruitment and retention from a wider range of backgrounds, including academic background, work experiences and hobbies. This would fit the increasingly high-profile, complex and ethically essential testing roles required by industry and society.

Competing interests

No competing interest is declared.

Author contributions statement

IE conceived the work, and designed the survey, CP and MM reviewed the survey design.
IE conducted the survey, IE, CP and MM analysed the results.
IE, CP and MM wrote and reviewed the manuscript.

Acknowledgments

The authors thank the participants their generous contributions of stories, the industry conferences and online community who helped to publicise the survey, and commented on early results. Thank you to the reviewers for their valuable suggestions.

References

1. T. Adlin and J. Pruitt. *The Essential Persona Lifecycle: Your Guide to Building and Using Personas.* Morgan Kaufmann, 2010.
2. B. Albert and T. Tullis. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics.* Interactive Technologies. Morgan Kaufmann, 2010.
3. Baris Ardic and Andy Zaidman. Hey teachers, teach those kids some software testing. In *2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG)*, pages 9–16. IEEE, 2023.
4. Robert D. Austin and Gary P. Pisano. Neurodiversity as a competitive advantage: Why you should embrace it in your workforce, 2017. Available from: https://hbr.org/2017/05/neurodiversity-as-a-competitive-advantage Accessed: (9 January 2024).
5. J. Bach and M. Bolton. A context driven approach to automation in testing, 2016. Available from: https://www.satisfice.com/download/a-context-driven-approach-to-automation-in-testing Accessed: (27 February 2025).
6. Sebastian Baltes and Paul Ralph. Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering*, 27(4):94, 2022.
7. Boris Beizer. *Software system testing and quality assurance.* Van Nostrand Reinhold Co., New York, 1984.
8. Boris Beizer. *Black-Box Testing: Techniques for Functional Testing of Software and Systems.* Wiley, 1995.
9. M. Bolton. Testing vs. checking, 2009. Accessed (5 February 2025).
10. M. Bolton. Testing is..., 2014. Accessed (19 January 2021).
11. Daryl Camilleri, Mark Micallef, and Chris Porter. Investigating cognitive workload during comprehension and application tasks in software testing. In *SEKE*, pages 237–242, 2022.
12. Daryl Camilleri, Chris Porter, and Mark J Micallef. Assessing task difficulty in software testing using biometric measures. In *35th International BCS Human-Computer Interaction Conference*, pages 1–10. BCS Learning & Development, 2022.

13. Capgemini. World Quality Report 2021-22, 2022. Available from: https://www.sogeti.com/explore/reports/world-quality-report-2021-22/ Accessed: (3 October 2022).
14. Capgemini. World Quality Report 2024-25, 2024. Available from: https://www.capgemini.com/insights/research-library/world-quality-report-2024-25/ Accessed: (20 February 2025).
15. Luiz Fernando Capretz, Daniel Varona, and Arif Raza. Influence of personality types in software tasks choices. *Computers in Human behavior*, 52:373–378, 2015.
16. Jerry D Cavin. The role of human factors in veteran sqa training. *Procedia Manufacturing*, 3:1535–1542, 2015.
17. Rikke Friis Dam and Teo Yu Siang. Personas: A simple introduction, 03 2022. Available from: https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them Accessed: (19 September 2024).
18. EuroSTAR. Eurostar Media Kit 2022, 2022. Available from: https://conference.eurostarsoftwaretesting.com/expo/ Accessed: (3 October 2022).
19. Isabel Evans, Chris Porter, and Mark Micallef. Scared, frustrated and quietly proud: Testers' lived experience of tools and automation. In *European Conference on Cognitive Ergonomics 2021*, pages 1–7. ACM, 2021.
20. Isabel Evans, Chris Porter, Mark Micallef, and Julian Harty. Test tools: an illusion of usability? In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 392–397. IEEE, 2020.
21. Isabel Evans, Chris Porter, and Mark J Micallef. Breaking tester stereotypes: who is testing and why it matters. pages 115–126, 2024.
22. Raluca Florea and Viktoria Stray. Software tester, we want to hire you! an analysis of the demand for soft skills. In *International Conference on Agile Software Development*, pages 54–67. Springer, 2018.
23. Raluca Florea and Viktoria Stray. The skills that employers look for in software testers. *Software Quality Journal*, 27(4):1449–1479, 2019.
24. Raluca Florea and Viktoria Stray. A qualitative study of the background, skill acquisition, and learning preferences of software testers. In *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering*, pages 299–305, 2020.
25. SFIA Foundation. Skills framework for the information age, 2025. Available from https://sfia-online.org/en Accessed: (29 January 2025).
26. Thomas R. G. Green and Marian Petre. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.
27. Monique Hennink and Bonnie N. Kaiser. Sample sizes for saturation in qualitative research: A systematic review of empirical tests. *Social Science I& Medicine*, 292:114523, 2022. Available from: https://www.sciencedirect.com/science/article/pii/S0277953621008558.
28. Felienne Hermans. *The Programmer's Brain: What Every Programmer Needs to Know about Cognition.* Simon and Schuster, 2021.
29. Tabata Pérez Rentería Hernández and Nicola Marsden. Understanding software testers in the automotive industry a mixed-method case study. In *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, pages 305–314. IEEE, 2014.
30. Julia Himmelsbach, Stephanie Schwarz, Cornelia Gerdenitsch, Beatrix Wais-Zechmann, Jan Bobeth, and Manfred Tscheligi. Do we care about diversity in human computer interaction: a comprehensive content analysis on diversity dimensions in research. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–16, New York, NY, USA, 2019. ACM.
31. Jon Hoeksma. Turning doctors into coders. *BMJ*, 347, 2013. Available from: https://www.bmj.com/content/347/bmj.f5142 Accessed: (27 February 2025).
32. ISO. ISO/IEC/IEEE 29119-1: 2013 Software and systems engineering - Software testing - part 1: Concepts and definitions. Standard, International Organization for Standardization, Geneva, CH, 2013.
33. ISO. ISO 9241-11:2018 Ergonomics of human-system interaction - part 11: Usability: Definitions and concepts. standard, International Organization for Standardization, Geneva, CH, 2018.
34. ISTQB. International software testing qualifications board, 2023. Available from: https://www.istqb.org/, Accessed: (26 March 2023).
35. Brittany Johnson, Yoonki Song, Emerson Murphy-Hill, and Robert Bowdidge. Why don't software developers use static analysis tools to find bugs? In *2013 35th International Conference on Software Engineering (ICSE)*, pages 672–681, 2013.
36. Tanjila Kanij, Robert Merkel, and John Grundy. A preliminary survey of factors affecting software testers. In *2014 23rd Australian Software Engineering Conference*, pages 180–189. IEEE, 2014.
37. Tanjila Kanij, Robert Merkel, and John Grundy. An empirical investigation of personality traits of software testers. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 1–7. IEEE, 2015.
38. MABL. State of Testing in DevOps 2021, 2021. Available from: https://www.mabl.com/resources/2021-testing-in-devops-survey-report Accessed: (3 October 2022).
39. Jenna E McChesney, Tara S Behrend, and Alexander Glosenberg. Stereotypical descriptions of computer science career interests are not representative of many computer scientists. *Scientific Reports*, 12(1):1–9, 2022.
40. André N Meyer, Earl T Barr, Christian Bird, and Thomas Zimmermann. Today was a good day: The daily life of software developers. *IEEE Transactions on Software Engineering*, 47(5):863–880, 2019.
41. Mark Micallef, Chris Porter, and Andrea Borg. Do exploratory testers need formal training? an investigation using hci techniques. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 305–314, 2016.
42. Glenford J Myers, Tom Badgett, Todd M Thomas, and Corey Sandler. *The art of software testing*, volume 2. Wiley Online Library, 2004.
43. Christopher P Nemeth. *Human factors methods for design: Making systems human-centered.* CRC press, 2004.
44. Jakob Nielsen. Corporate UX Maturity: Stages 1-4, 2006. Available from: https://www.nngroup.com/articles/ux-maturity-stages-1-4 Accessed: (4 January 2021).

45. Don Norman and Jakob Nielsen. The definition of user experience (ux), 2019. Available from: https://www.nngroup.com/articles/definition-user-experience/ Accessed: (Feb. 8, 2021).
46. Marian Petre and André van der Hoek. *Software Design Decoded: 66 Ways Experts Think*. MIT Press, Cambridge, MA, 2016.
47. Practitest. State of testing report, 2022. Available from: https://www.practitest.com/state-of-testing/, Accessed: (3 October 2022).
48. Practitest. State of testing report, 2024. Available from: https://www.practitest.com/assets/pdf/stot-2024.pdf, Accessed: (20 February 2025).
49. H.C. Purchase. *Experimental Human-computer Interaction: A Practical Guide with Visual Examples*. Cambridge University Press, 2012.
50. C. Righi and J. James. *User-Centered Design Stories: Real-World UCD Case Studies*. Interactive Technologies. Morgan Kaufmann, 2010.
51. Gema Rodríguez-Pérez, Reza Nadri, and Meiyappan Nagappan. Perceived diversity in software engineering: a systematic literature review. *Empirical Software Engineering*, 26:1–38, 2021.
52. J. Sauro and J.R. Lewis. *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann, 2016.
53. Ben Shneiderman. Artificial intelligence needs both pragmatists and blue-sky visionaries. urlhttps://www.scientificamerican.com/article/artificial-intelligence-needs-both-pragmatists-and-blue-sky-visionaries/ Accessed: (7 October 2022), 2022.
54. Janice Singer and Norman G. Vinson. Ethical issues in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 28(12):1171–1180, 2002.
55. Justin Smith, Lisa Nguyen Quang Do, and Emerson Murphy-Hill. Why can't Johnny fix vulnerabilities: A usability evaluation of static analysis tools for security. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 221–238. USENIX Association, August 2020. Available from: https://www.usenix.org/conference/soups2020/presentation/smith.
56. Viktoria Stray, Raluca Florea, and Lucas Paruch. Exploring human factors of the agile software tester. *Software Quality Journal*, 30(2):455–481, 2022.
57. Techwell. Techwell Media Kit 2022, 2022. Available from: https://www.techwell.com/advertise-with-techwell/media-kit-form Accessed: (3 October 2022).
58. Aleksis Tulonen. Project Mercury and the Death of Testing, 2013. Available from: https://flowoftesting.wordpress.com/2013/12/21/project-mercury-and-death-of-testing/ Accessed: (26 February 2025).
59. Wijnand A P van Tilburg, Eric R Igou, and Mehr Panjwani. Boring people: Stereotype characteristics, interpersonal attributions, and social reactions. *Personality & social psychology bulletin*, 25:1461672221079104–1461672221079104, 2022.
60. Vanessa Vella and Chris Porter. Wait a second! assessing the impact of different desktop push notification types on software developers. In *Proceedings of the 33rd European Conference on Cognitive Ergonomics*, pages 1–7, 2022.
61. Pradeep Waychal, Luiz Fernando Capretz, Jingdong Jia, Daniel Varona, and Yadira Lizama. Practitioners' testimonials about software testing. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 582–589, Washington, DC, USA, 2021. IEEE, IEEE Computer Society.
62. Erica Weilemann. A winning team-what personality has to do with software engineering. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 252–253. IEEE, 2019.
63. Bernhard Wöckl, Ulcay Yildizoglu, Isabella Buber, Belinda Aparicio Diaz, Ernst Kruijff, and Manfred Tscheligi. Basic Senior Personas: A Representative Design Tool Covering the Spectrum of European Older Adults. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, page 25–32, New York, NY, USA, 2012. Association for Computing Machinery.

Isabel Evans Isabel Evans is a postgraduate research student in the Computer Information Systems Department at the Faculty of ICT, University of Malta. Her research focuses on Human Factors in Software Testing. Isabel works in industry, specialising in testing and software quality from 1985 onwards.

Chris Porter Chris Porter is an Associate Professor in the Computer Information Systems Department at the Faculty of ICT, University of Malta. He holds a PhD in Computer Science from University College London, and his research focuses on Human-Computer Interaction and Human Factors, with a particular interest in web accessibility, assistive technologies, and software engineering activities. Chris also manages the HCI Lab at the Faculty of ICT and leads the Interaction Design Foundation (IxDF) Malta group.

Mark Micallef Mark Micallef is an Associate Professor in the Computer Science Department at the Faculty of ICT, University of Malta. He holds a PhD in Computer Science from University of Malta, and his research focuses on open problems in software testing with a particular focus on the human tester, and AI.

Survey Questions

(Instructions and information for Participants)

Informed Consent

Consent::I have read and understood the above statements and agree to participate in this study. (yes/no)

Section 1

Demographics and Background

1. How did you hear about this survey (which conference, event, article, or by email, etc)?
2. Just to get you started, and also to give us a rounded view of who testers are.... What are your hobbies and pastimes outside work?
3. What is your age?
4. What is your gender?
5. Where do you live?
6. What is your highest academic qualification?
7. What professional qualifications do you have?
8. What training in software testing have you had?
9. Where do you mainly work (your office)
10. What is your employment status? (check all that apply) (full-time, part-time, seeking work, retired)
11. How many years have you worked on software projects and in what roles? (for each of the roles below, tabulated, with ranges of years under 2 years, 2-4 years, 5-10 years, 11-20 years, over 20 years) Requirements analysis

- System architecture
- System analysis
- System design
- UX design
- Development
- Software testing
- Test Management
- Test Programme Management
- Release management
- Operations
- Support
- Product ownership
- Risk management
- Project and programme management
- Agile team leadership
- Coaching
- People management
- Other responsibilities (enlarge in the next question)

12. Please tell us any additional relevant information about your experiences and roles What is your job title
13. Does your job title reflect your actual work (Scale 1= not at all, 10 = completely)
14. Please elaborate on your answer about your job title:
15. What are your work and career aspirations?
16. You have now completed the first section. Thank you! The next section is about what you do. Are you ready to proceed? (yes, no - use my answers so far, no - please delete my answers)

Section 2

Working Environment

17. Tell us a little about the commercial basis for the software you test (e.g. commercial, built for in-house use, freeware, open-source, etc)
18. What is the main domain where the software is used (e.g. medical, financial, office use, avionics, gaming, etc)
19. What background and experience do you have in the software domain you are currently working on
20. What testing or other IT activities are included in your current role (job description, a terms of reference, a contract, the job advert...) NB: this might be a paid role, or a voluntary project or both.
21. What else do you do - either testing or other IT activities - *in addition* to your current role as described above
22. What testing or other IT activities *don't* you do that are in your current role as described above
23. You have now completed the second section. Thank you! The next section is about approaches, techniques and tools. Are you ready to proceed? (yes, no - use my answers so far, no - please delete my answers)

Section 3

Approaches, techniques and tools

24. Broadly, for the testing related tasks and activities you do, what approaches do you take? Please answer with the task/activity followed by the approach(es).
25. What techniques do you most usually use to support your testing related tasks and activities? Please answer with the task/activity followed by the technique(s).
26. What tools do you do you most usually use to support your testing related tasks and activities? Please answer with the task/activity followed by the tool(s)
27. Who chooses the tools and techniques you use for testing-related activities and tasks?
28. What limitations and constraints do your current tools place on you, that prevent you doing great work?
29. Thinking about improvements to the tool support for your testing tasks and activities, suggest the 1 or 2 most important improvements to tools that you'd welcome?
30. You have now completed the third section. Only one more short section which is a chance for you to add anything else you think useful... Are you ready to proceed? (yes, no - use my answers so far, no - please delete my answers)

Section 4

What else?

31. What else should we have asked you...?
32. And... what are your answers to your questions...?
33. Would you like to take part in follow up activities for this research? For example, a follow up interview, take part in case study or further surveys, receive copies of academic papers.

If so: Your name - Your email address -

(End of survey.)