

Stephan Goericke *Editor*

# The Future of Software Quality Assurance



# Next-Generation Software Testers: Broaden or Specialize!



**Erik van Veenendaal**

**Abstract** Software has become immensely important for today’s society. Despite many quality initiatives, the IT-industry is still far from being able to deliver zero-defect software. In recent years the way software is being developed has changed dramatically in most world regions. In addition to the rapid and dynamic changes currently in the software development arena, there is an increased growth in innovation, new technologies, and expansion of IT throughout most industries. There has been a large shift towards adopting an Agile and/or DevOps way of working. Agile typically provides benefits such as the ability to better manage changing priorities, improved project status visibility, increased team productivity and better delivery predictability. However, many organization are struggling with Agile and scaling Agile, and it has become apparent that moving towards Agile does not automatically also guarantee improved software quality. Testing, although in Agile organized differently compared to traditional organizations, is still and will remain an important part of software development. This is not only due to the importance of software in today’s society, but also due to the many (technical) challenges that IT project are facing, e.g., increasing complexity, new technologies, systems-of-systems, variety of devices and OS’s and security vulnerabilities. What does all of this mean for the software tester, and to the knowledge and skill set that is expected of a tester? This chapter will look in detail at the knowledge and skill set required for a tester to add value and survive in the rapidly changing IT world. Two options will be provided to the tester:

- Broaden your skill set and become a so-called T-shaped tester.
- Choose a test specialization, but choose this with great care and a vision towards the future.

**Keywords** Software testing · Software quality · Software tester · Software testing skills

---

E. van Veenendaal  
TMMi Foundation, Dublin, Ireland

## 1 The Future of Testing

Before looking at answers and solutions regarding the knowledge and skill set required for a tester, let's first briefly look at where we are today and what the future of the testing profession looks like (if at all possible). Most people perceive Agile as something that is the common way of working around the globe. However, there are also countries where the number of people or organizations actually practicing Agile are still a minority. Perhaps these countries are just "behind" and they will be Agile a few years from today, but it can also be the case that Agile does not always have a perfect fit within every culture.

There are keynotes at international testing conferences claiming that testers will soon disappear. According to them, there will be no more, or at least very few, dedicated testers in the near future. Interestingly, numbers from survey reports, e.g. the World Quality Report, show exactly the opposite. Also test certification schemes such as ISTQB® and TMMi® are still showing a strong growth and uptake. Surprisingly contradicting talks, signals and facts. Which of them are true, and what is the future for the software testing profession?

In most western societies Agile seems to be a good fit. Generally speaking, there are many people who tend to be communicative, liberal, open minded, love working in a team and less focused on formalities. Mind you, not all parts of the world and cultures are like this. Even within Europe there are huge differences, sometime even between neighbouring countries. Agile has shown to deliver many advantages over the years, but also in a traditional environment delivering a quality product is certainly not impossible.

With all of these opposite and different trends it is quite difficult to accurately predict the future of the testing profession. However, I strongly believe it is safe to state that with the current state of the practice in terms of software quality being delivered and its criticality, the need for testing as a discipline and as a profession will remain (at least for short to medium term). Looking at testing today and tomorrow, there is a firm tendency towards two main options for the tester:

- Become a so-called *T-shaped person* (tester), by changing your attitude and by *broadening* your knowledge and skills. Knowledge and skills will be a challenge in the near future for many testers. It is just not good enough anymore to understand testing and hold an ISTQB certificate. Testers will most often not work in their safe independent test team environment anymore. They will work more closely together with business representatives and developers helping each other when needed and as a team trying to build a quality product. Besides strong soft skills, it is also expected from testers to have knowledge of the business domain, requirements engineering, scripting, etc. Become a "tester plus", someone who can test, but also organize testing and support others in testing. A tester that can do much more than just test.
- Become a *test specialist*. As products are becoming more and more complex, and are integrated in an almost open environment, many so-called non-functional testing issues have become extremely challenging. At the same time the business,

users and customers do not want to compromise on quality. To be able to still test non-functional characteristics such as security, interoperability, performance and reliability, or other complex aspects, e.g. systems-of-systems, highly specialized testers will be needed. Even more so than today, these specialists will be full-time test professionals with in-depth knowledge and skills in one specific (non-functional) testing area only.

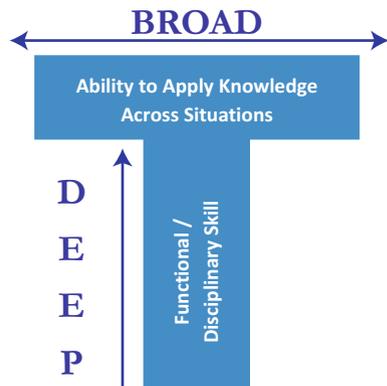
The concept of a T-shaped person is of course popular in the Agile world and refers to the need for cross-skilled developers, business analysts and testers in an Agile team, e.g. a Scrum team. In practice, many talk about being a T-shaped tester, but only a few truly are. Let’s try to define the knowledge and skill set required to be a true T-shaped tester, but before we do let’s look into more detail on what the concept of T-shaped persons actually means and stands for.

## 2 The Concept of T-Shape

The concept of T-shaped skills, or T-shaped persons, is a metaphor originally used in job recruitment to describe the abilities of persons in the workforce. The vertical bar on the T represents the depth of related skills and expertise in a single field, whereas the horizontal bar is the ability to collaborate across disciplines with experts in other areas and to apply knowledge in areas of expertise other than one’s own (see Fig. 1). More in detail the horizontal stroke is composed of two things. First, empathy. It’s important because it allows people to look at a problem from another perspective—to stand in somebody else’s shoes. Second, they tend to get enthusiastic about other people’s disciplines, to the point that they may actually start to practice them. T-shaped people have both depth and breadth in their skills.

To better understand what a T-shaped person is, it is perhaps easier to first understand what the converse, a so-called I-shaped person, is. An I-shaped person is one who is a functional expert—their functional expertise being represented by the

Fig. 1 T-shaped person



vertical stroke in the letter I. There is of course nothing wrong in being an I-shaped person—a functional expert. However, let’s imagine a number of functional experts trying to work together on a new mobile app. An app developer, an SEO expert, an analytics expert, a content developer, and an art director have a kick-off meeting to decide on a strategy for the new mobile app.

The SEO expert insists that you should build the app around a keyword map to make sure that the site’s structure mirrors an emphasis on keywords. The app developer insists that the mobile app be as easy to code as possible. The analytics expert says that the new design has to be based on what the app analytics show about usage of the current app. The content developer insists it’s all about developing interesting, engaging navigable content. And finally, the art director is insisting that app composition and brand beauty is the number one goal. Which one of these I-shaped people is right? How do we manage all these different opinions and make decisions? No matter how good the I-shaped functional experts are at their individual functions, what they lack is not only an appreciation of their co-workers’ areas of expertise, but also the training to actually find solutions at the intersection of their respective functional areas.

Let’s now compare the I-shaped persons to those being T-shaped. A T-shaped person is typically multi-function aware, collaborative and seeking to learn more about how their function impacts others and the end product. T-shaped people are far more flexible, and more able to easily catch on to new trends, but are of course not as substantial in each adjacent discipline as in their primary skill. Contrary to the I-shaped person, the T-shaped specialist tend to get the general picture rather than immerse themselves in details, unless it’s really needed.

In addition to I- and T-shaped concepts, there are descriptive variations that have emerged recently, the most common of which are (see Fig. 2):

- $\pi$ -shaped—two legs of key skills connected with the dash of general knowledge
- M-shaped—three (or more) key, deep skills

Although the concept of going beyond T-shaped, such as  $\pi$ -shaped or even M-shaped is certainly an interesting one for some disciplines, it probably is not the way to go for testers. As we have learned over the years a certain degree of

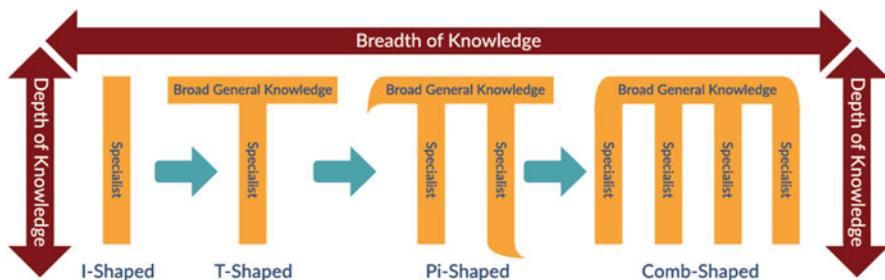


Fig. 2 Variations to the T-shaped concept

independence most often makes the tester more effective at finding defects due to differences between the author's and the tester's cognitive biases (critical distance). Having multiple specialties by being a  $\pi$ -shaped or even M-shaped tester, would typically make it much harder to keep the independent perspective. In these cases, you would as an expert be involved in tasks that you at the same time as a tester should evaluate. In Agile, preserving independence is already often more difficult when a tester is embedded in a team. At the same time, also being a  $\pi$ -shaped person possessing (and performing) another specialty beyond testing would probably make the required level of independence almost disappear altogether.

T-shaped people and the teams they work in can achieve results far better than teams that consist of only I-shaped people. But the development of T-shaped people is a serious, long-term undertaking and most often largely underestimated. It requires people with the right attitude and self-determination to start, but then it requires effort to continue to provide them with the training and resources they need and the type of safe collaborative environment that allows for T-shaped person and teams to perform at their best.

### 3 The T-Shaped Tester

To drive a career in software testing, what are the most valuable knowledge and skills to acquire. As already stated, one way to develop a testing career is to specialize by going deeper in a single specific non-functional or niche. These kinds of specialists are regarded in theory as being I-shaped, which means that their skills are seen as being very narrow but extremely deep. However, in a fast-paced world, this strategy has evident risks, such as if the area of specialization becomes outdated, unpopular or, as we will see later, when a specialized area changes to one that becomes common to all.

While in previous decades there was a demand for I-shaped testers, there is now a growing opportunity for T-shaped persons because those who have deep skill in one discipline and in addition general knowledge across disciplines will much easier be able to work in a changing environment. In the Agile world, the T-shaped tester is a team member whose key expertise is testing but who can also provide support in other activities, for example, those that lie in the fields of programming or business analysis (requirements engineering). So, in relation to T-shaped, we should look for the skills that can potentially boost the testers' profile. For a professional software tester, good options would be:

- *Testing*: have a deep and broad knowledge across the testing domain
- *Other development specialties*: business analysis, programming, technical writing, etc.
- *Domain knowledge*: Medicine, Insurance, Banking, IoT, etc.
- *Soft skills*: they have a positive impact on personal effectiveness, leadership and collaboration with others

Discussing the skill sets of T-shaped testers, we should also be aware of the proportions between ‘horizontal’ and ‘vertical’ aspects in the skill set. Depending on the environment, the need in each family of skills will differ. Those who have very deep and narrow expertise in the field can become over-skilled, as employers don’t tend to pay for skills they don’t need. Those who have broader skills can feel the lack of expertise in their key discipline at some point and will need to catch up on it in the short term if required. Hereafter the four knowledge and skill options identified for the T-shaped are elaborated upon with examples.

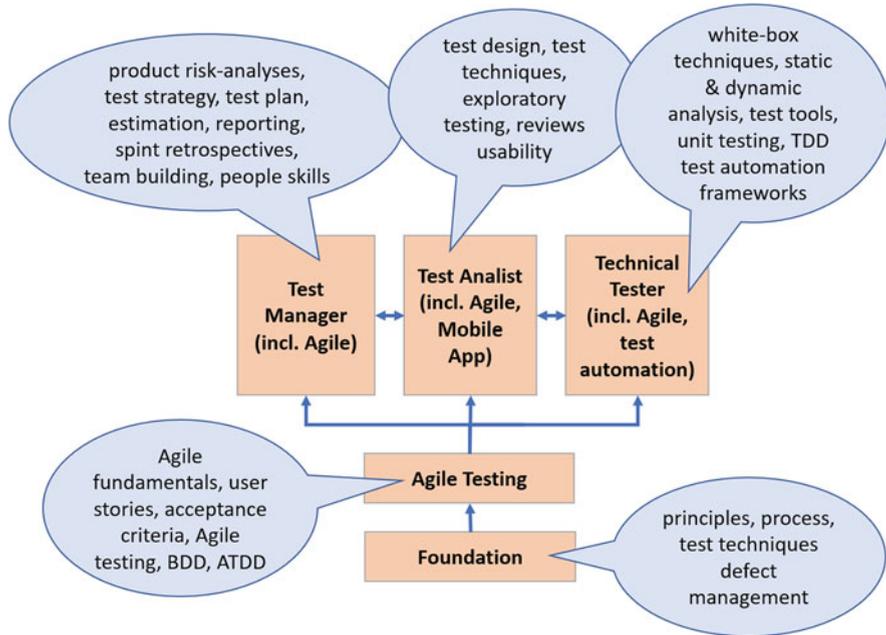
### ***3.1 Deepening: Testing Knowledge and Skills***

Today’s tester needs to have a full toolbox with varying test methods and techniques. Working in a team, depending on the context and charter, the most appropriate methods and techniques shall be selected from the toolbox. Trying to define the toolbox for the tester, and thus the testing knowledge and skills required, the ISTQB product portfolio can be used as a reference model. Although there is much criticism on ISTQB in some testing communities, from a content point-of-view there is without doubt much interesting knowledge and material available across many areas of testing and documented in the various syllabus. ISTQB today is much, much more than the basic ISTQB foundation level syllabus.

If we take the ISTQB product portfolio as a starting point (see [www.istqb.org](http://www.istqb.org) for the latest version of the portfolio), many interesting topics and syllabi are available. Trying to define the required testing knowledge and skills, it is at least strange to see that ISTQB does not consider Agile testing to be a part of the core set of knowledge and skills. It is defined within the portfolio as a separate stream. Also interesting to see is that test automation and mobile application testing are considered as specialist areas within testing. Today, these are almost like standard requirements for a tester. The fact that these were originally defined as specialist areas by ISTQB perhaps shows how quickly the market changes. What is defined as a specialist area today could well be a common requirement for knowledge and skills tomorrow.

The picture in Fig. 3 is by no means intended to be complete or based on some extensive survey. It is intended to show on a high level what is expected from a tester in terms of testing knowledge and skills today and for sure tomorrow.

Having attended an ISTQB Foundation course (and having passed the exam) and then stating you are a professional tester ready for the future is almost like a joke. A 3-day ISTQB Foundation course is based on a well-founded syllabus, but it only teaches the basics and principles of testing and doesn’t get the job done. One will also need to be trained in Agile testing, which I believe is core and not optional for any tester. The real meat is in attending more advanced hands-on, practical courses or workshops in which you will learn how to actually apply testing practices in various contexts. These advanced courses should include areas such as test automation and mobile application testing, and of course should be taught from an Agile perspective. Following the T-shaped concept and having a deep



**Fig. 3** Tester knowledge and skill set

knowledge in testing, we do not expect testers to choose between follow-up areas such as test manager, test analyst or technical test analyst. We expect testers to cover all three areas and become a true professional. As an example, there are fewer and fewer dedicated test managers or test leads. Many testers today are embedded in an Agile team, as such they perform testing tasks, but also coach and support business analysts doing functional user-story-based testing and developers doing automated unit testing. Being a tester in an Agile team also means you are involved in tasks that were originally in the exclusive domain of the test manager, e.g. product risk sessions, estimations, retrospectives, reporting, etc.

Note there are many other means of acquiring the same testing knowledge and skills; ISTQB is just one option and used here as an example. There are often great tutorials at testing conferences that discuss interesting topics and areas. There are also many highly practical courses available in the market, e.g. on test automation. Of course alternatives exist to doing formal training as well, e.g. mentoring, on-the-job learning, etc.

**Not Just the Tester!**

There will most likely be professional testers in the future, but even more so testing as an activity will remain to be extremely important and challenging. Not only the tester performs testing activity, also other team members, e.g. developers and business analysts, will perform testing activities especially as a result of Agile

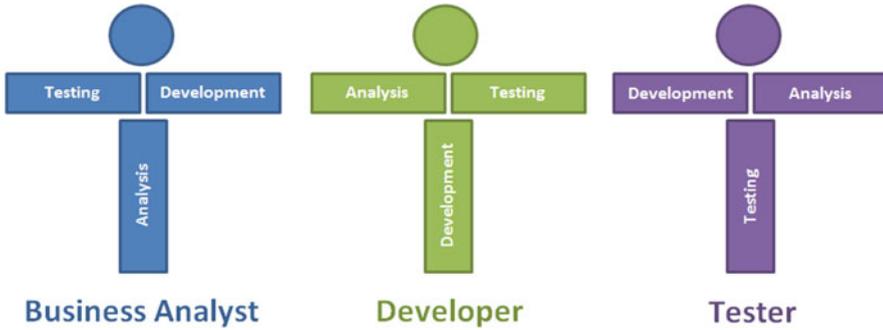


Fig. 4 T-shaped Agile team

transformation. Following the T-shaped concept, the tester is expected to have deep knowledge and skills in testing, but from other team members it is expected that they acquire testing knowledge and skills as part of their horizontal skill set bar (see Fig. 4). We cannot just direct other people to be involved in testing, they need to be trained and coached in testing to be able to perform this task. Remember what Glenford Myers stated in his book *The Art of Software Testing*: “Testing is an extremely complex and intellectually challenging task”.

### 3.2 Broadening: IT Knowledge and Skills

Working in a cross-functional team closely with developers and business analysts implies that at least a tester needs to appreciate and understand what other team members are doing and preferably also be able to support them in their tasks. From a T-shaped tester, it is typically expected they can support a software developer with unit testing and a business analyst when they are defining acceptance criteria for user stories. It is required to understand the life cycle model that is being used and the technical environment in which development takes place. IT knowledge and skills for a tester should cover a wide and varying range of knowledge skills of which some essential examples are listed and described hereafter.

#### Requirements Engineering/Business Analysis

Testers are of course one of the main stakeholders. Test planning, product risk analysis, test cases are all based upon requirements. Testers are involved in requirements reviews, and need to understand what level of quality is reasonable to ask for. Often in Agile teams, testers support the identification and specification of requirements (user stories) and their acceptance criteria.

#### Programming

It isn't that testers need to work like a developer, but it is important to understand the inside of the application so that it becomes easy to comprehend its functioning

and risks areas, and create tests accordingly. Programming knowledge helps in identifying possible errors in the code and work closely with the developer on static analysis and unit testing, possibly using Test Driven Development. It is advisable to learn at least two programming languages, e.g. Python, Java or C++. Of course having programming skills also strongly support the ability to perform test automation tasks. With the increasing complexities and integrations in the application, relying on manual testing alone cannot get the job done.

### **Web and Mobile Technologies**

Today most testers must also become familiar with the web and mobile technologies so that they can understand the application, its build and scalability and apply a suitable course of actions for its testing. It is highly important that testers keep an eye open on web and mobile technology advancements since it guides them in comprehending the coding architecture and technical challenges to deliver effective testing solutions.

### **Software Development Lifecycle**

Testers need to learn and understand the software life cycle as it will help them understand the development tasks and plan testing cycles accordingly. Having an in-depth knowledge of software life cycle will also help anticipate challenges in the development process which can guide in taking the right measures beforehand. With Agile and DevOps methodologies being popular, testers of course need to learn and understand these as well, especially the impact this has on how testing is performed.

### **Project Management**

Learning the skills of project management will support the tester in becoming a better test manager. Project management skills also prepare testers to be accountable and answerable for their work to concerned stakeholders and also undertake responsibility and management for specific test activities. This way, project management skills contribute to delivering quality results, improving the entire test process.

### **Configuration Management**

The purpose of configuration management is to establish and maintain the integrity of the component or system, the testware and their relationships to one another. For the tester this an important process to ensure quality and as such it is essential to have some knowledge on configuration management. In addition also, testware needs to be managed via this process. All items of testware should be uniquely identified, version controlled and tracked for changes, related to each other and related to versions of the requirements so that traceability can be maintained.

## ***3.3 Broadening: Domain Knowledge***

In this context, domain knowledge is defined as knowledge about the environment in which the target system operates. For a tester it's important to understand the domain in order to be able to communicate with business stakeholders (product

owners), but also to make the right decisions while performing testing activities. Remember, exhaustive testing isn't possible, and testers are making trade-off decisions all the time, and they'd better be the right ones! What features are most important to test, which configurations occur most often, etc.? Hence for a tester possessing domain knowledge along with the [other skills](#) is a big plus to the industry. In the context of being a T-shaped tester, there are also benefits outside of testing. A tester with domain knowledge can probably partly relieve a business analyst, or assist other team members by bringing the necessary domain perspective.

Some examples where a tester will benefit from having domain knowledge are discussed next.

### **Understanding Risks and Better Test Cases**

Unless you are aware of the domain, you can't identify and analyse the product risks, and write and execute test cases to effectively simulate the end user. It's not just about using at a particular activity, it is required throughout all testing activities.

### **Understanding Impact**

When an issue arises, a tester understanding purpose of the functionality of the system will much better be able to analyse the impact of the issue. For example, when a defect is found in the payment process of an online pizza order application, a domain-based testers will have a clear idea about the process steps impacted needed for a successful transaction. This will assist the tester in doing better confirmation and regression testing when the defect is fixed.

### **More Important Defects**

Domain knowledge testers are high in demand due to their ability to understand the application beyond just [finding defects](#), e.g. during exploratory testing. They typically find more important defects.

### **Prioritize Defects**

Since the tester understands the domain, the tester will have a clear idea of how to best prioritize the outstanding set of defect fixes.

### **More Effective Reviewing**

A tester with domain knowledge can be more productive at the start of the project or iteration. Good knowledge of the functional flow of the business processes and business rules will help better understanding the requirements and as a consequence be able to perform reviews more effectively.

Being in an IT-dominated world, the value of a tester with domain knowledge is incredible as it is undoubtedly a critical success factor for testers. While testing any application, it is important to be able to think from an end user's perspective since they are the ones who are going to use the product. Domain knowledge usually must be learned from end users (as domain specialists/experts) and may, amongst others, include user profiles, workflows, business processes, business policies and configurations. Without going into detail on how to acquire business and domain knowledge, there is of course much more than just attending a training, also consider

apprenticing, observing users/customers actually using the application, visiting online forums and becoming part of communities.

Note, there seems to be a tendency to prefer technical testers in the Agile community, but as we have learned in this paragraph, there is a strong need and benefit to having testers with domain knowledge and sometimes even a domain background, e.g. end users that have become testers! Of course there never is a right answer in these kind of situations, but it is something that needs to be considered and balanced when assigning testers to a team and defining a required knowledge and skill set for the T-shaped tester.

### **3.4 Broadening: Soft Skills**

Any software tester should also possess so-called soft skills (also known as people skills). Soft skills are important as they are used to approach work. Testers have an instinct and understanding for where and how software might fail, and how to find defects. A tester also should have the soft skills to influence and communicate in a manner that they become vital to the project. Testing requires a toolbox full of soft skills, including communication, time management, analytical skills, eagerness to learn and critical thinking, but also relatively standard people skills such as reading, reporting and presentation skills. Some of these skills allow a tester to be better in finding defects, but most of them relate to being better at communicating a, most often difficult, message. Having these soft skills is like a prerequisite to having the right attitude for being a T-shaped, open-minded tester within an Agile team, e.g. have empathy towards other disciplines, knowledge sharing and being a team player. Being a T-shaped tester and being part of a team means the soft skills that the tester possesses can also be used to support other team members at their tasks.

Hereafter, some of the important soft skills which a tester should possess in order to excel in his field are briefly described.

#### **Communication**

Since a tester has to deal with so many different team members, it is very important to have a proper communication channel with them. Whether it is a defect to report, explanation or clarification on an ambiguous requirement, a tester has to communicate with respective business analysts, developers and sometimes end users. When you have good communication skills, you eliminate ambiguity and misunderstanding while talking to the different team members. Also, most of the issues which arise due to a communication gap would be at par. For example, when you find a crucial defect, it is very important to explain it in a polite way so that the developer doesn't feel like you are blaming him for the root cause.

#### **Time Management**

Time management is the most important skill for a tester, even in Agile the last days on an iteration are usually packed with "stressful" last minute testing. When you know how to use your time properly and how to prioritize tasks according to the end

date, you will end up meeting the end dates more easily and encounter less work pressure.

### **Analytical and Detail Oriented**

This is one of the skills which can help you find more and the most important defects. When you understand and are able to analyse requirements not only gaps in the requirements can be found and fixed but also you get to know the whole functionality, flow and the expected outcome of the product. Sometimes details could be missed that later result in a major issue which are much more difficult to rework. To mitigate such situations it is also very important to have a 'detail-oriented attitude'.

### **Eager to Learn**

If you're eager to learn, and willing to take on activities outside your comfort zone needed to help the team deliver a good product, you have a bright future as a tester. It is crucial for the software tester to be aware of the latest tools and technologies and to keep learning without fail. When you have knowledge about the latest tools and technologies, you make better decisions.

### **Critical Thinking**

Critical thinking is the kind of thinking that specifically looks for problems and mistakes. It is the ability to reason by carefully analysing something in order to determine its validity or accuracy. Critical thinking is about being an active learner rather than a passive recipient of information. It is possibly the most important type of thinking in the context of testing. As testers, we should always question ideas and assumptions rather than accept them at face value.

### **Knowledge Sharing**

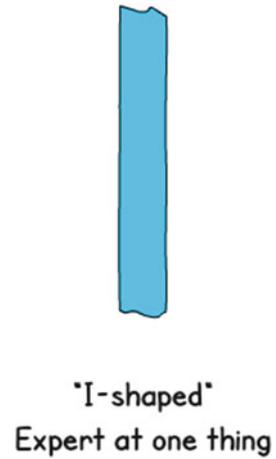
A tester should have an attitude of helping and coaching his Agile team members by sharing his knowledge. Not only will this avoid gaps and confusion, allow them to assist in and do better testing themselves but also motivate other team members to share their expertise and knowledge.

### **Team Player**

Being part of an Agile team, you always need a team effort to achieve something. A tester motivates the team towards better testing and good levels of product quality. To achieve quality software, it is important for a software tester to be a good team player.

## **4 The Test Specialist**

As already stated in the introduction paragraph, there is an alternative to the career path of becoming a T-shaped tester: become a test specialist. A specialist is defined as a person who concentrates primarily on a particular subject or activity; a person that is highly skilled in a specific and restricted field. Following the concept of

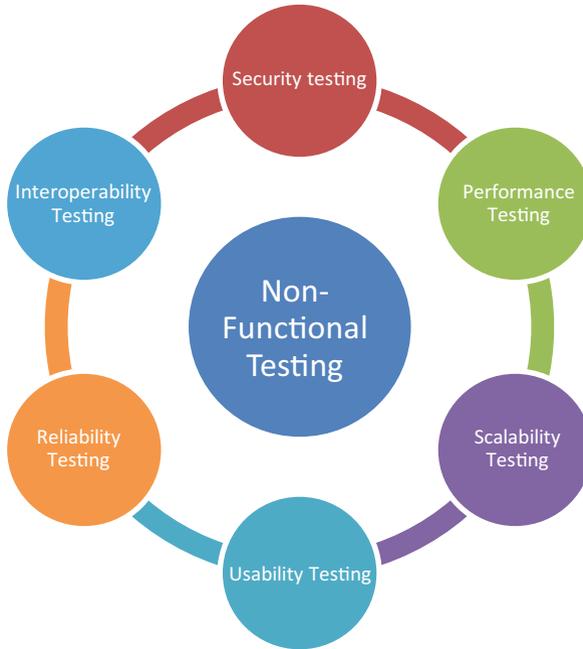
**Fig. 5** Test specialist

“shaped” persons, a test specialist is an I-shaped person (see Fig. 5). A tester with a deep (vertical) expertise in one testing area and less knowledge and skills in other testing areas, let it be outside testing. Their expertise in that one testing area is much, much deeper than the expertise possessed by a T-shaped tester for the same testing area.

As products are becoming more and more complex, and are integrated in an almost open environment, many so-called non-functional testing issues have become extremely challenging. You cannot just do this on the side, this requires much specialized knowledge and skills, training and dedication. To some testers it may not be their piece of cake, or may even be too (technically) difficult to master. As a result, to still be able to test an IT-industry non-functional characteristics such as security, interoperability, performance and reliability, or other complex aspects like systems-of-systems, highly specialized testers will be needed. These specialists are typically full-time test professionals with in-depth knowledge and skills in one specific (non-functional) testing area only.

Also from a customer point of view that comes to us for a solution, the customer may sometimes come to the door with a problem that is to be addressed by a single specialist only. It doesn't require a team-based solution. It is with these kinds of problems or questions that the I-shaped specialist clearly has much added value.

So what are typical test specialists areas? Again, let's look at the ISTQB product portfolio as a reference framework to identify some testing topics that are considered to be areas where we would benefit from having test specialists. ISTQB clearly points us in the direction of non-functionals (see Fig. 6), e.g. security test specialist, usability test specialist or performance test specialist. These are valuable test specialists in the context as defined above. ISTQB also mentions amongst others automotive software tester, gambling industry tester and model-based tester. Personally I'm not sure these are test specialist areas that contain and require enough specialist knowledge and skills to be able to add value and survive as an I-shaped



**Fig. 6** Examples non-functional test types

tester. Probably these areas are more specific to a certain domain and belong in the knowledge and skill set of a T-shaped tester in that domain.

Finally, it is also interesting to see that ISTQB also considers test automation and mobile application testing to be specialist areas within testing. Today, these are almost like standard requirements for a tester. The fact that these were originally defined as specialist areas by ISTQB perhaps shows how quickly the market changes. What is defined as a specialist area today could well be a common requirement for knowledge and skills tomorrow. This also points out the danger of being an I-shaped test specialist. Today, there is a huge demand for security test specialists and perhaps slightly less for performance test specialists. However, this may rapidly change or the specialism gradually moves to become a generalized knowledge and skill area that is required for T-shaped testers. Usability testing is probably somewhere in that transition.

To summarize, test specialists are needed and have much added value. It certainly is an alternative from being a T-shaped tester. Be careful which specialism to choose and keep an eye out for what is happening in the market today and the near future. Sometimes is it possible to hop from one test specialism to a new one when the current one becomes obsolete. At the same time, it doesn't hurt to also have some knowledge and skills that typically belong to the T-shaped tester as a fallback scenario. The latter will also be beneficial when working less stand-alone and being

part of a team. It will again also help to appreciate what others are doing and assist in looking at things from a different perspective.

## 5 Conclusions

With the current state of practice of the IT industry, we are far from achieving zero defects (if at all in the future). Software testing is and will remain an indispensable part of software development. The required knowledge and skills of the test professional of the (near) future are discussed. In detail, the T-shaped tester was presented and explained. In practice many talk about being a T-shaped tester, but I believe in reality we are far from being there. The need for I-shaped testers was also briefly presented. There are basically two options to choose from:

- Broaden your knowledge and skills and become a true test professional (T-shaped tester)
- Deepen your knowledge and skills in a specific testing area and become a test specialist (I-shaped)

For testers driving their career, it is extremely important to define individual direction of growth and development. Not forgetting the T-shaped concept, base your choice on your own strengths and passion, and take into account your work environment (life cycle, domain), trends in the industry and the current (and future) demands of the job market. In addition, reviewing your [career plan](#) on a regular basis will help you stay on top, and get the best value out of the work you do.

**Acknowledgement** An explicit thank-you is to Bob van de Burgt, who reviewed this book contribution in detail and provided valuable feedback.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

