

The Tester Skills Program



Teaching Testers to Think for Themselves

Paul Gerrard

Abstract In 2018, Gerrard Consulting was approached by the IT@Cork Skillnet (<https://www.skillnetireland.ie>; <https://www.itcork.ie>) and Softtest (<https://softtest.ie>) organizations and asked to support an initiative that aimed to improve the skills of testers in the south of Ireland software community.

Around 20 testing and QA managers had decided to look at their challenges and plan a way forward to develop and improve the testing-related training available to local testers. The first meeting took place on 29 November 2018. The presentation that introduced the initiative was titled, ‘Developing Testing Skills to Address Product Quality Challenges’.

What started as an attempt to create a 3-day class for beginner testers became a much more substantial learning and development (L&D) program. This chapter describes the reasons why the program is necessary, the current status and how it is likely to evolve in the future.

Keywords Software testing · Software quality · Software testing skills · ISTQB · Software tester

1 Introduction

1.1 Background

In 2018, Gerrard Consulting was approached by the IT@Cork Skillnet [1, 2] and Softtest [3] organizations and asked to support an initiative that aimed to improve the skills of testers in the south of Ireland software community.

Around 20 testing and QA managers had decided to look at their challenges and plan a way forward to develop and improve the testing-related training available to local testers. The first meeting took place on 29 November 2018. The presentation

P. Gerrard
Gerrard Consulting, Macclesfield, UK

that introduced the initiative was titled, ‘Developing Testing Skills to Address Product Quality Challenges’.

What started as an attempt to create a 3-day class for beginner testers became a much more substantial learning and development (L&D) program. This chapter describes the reasons why the program is necessary, the current status and how it is likely to evolve in the future.

1.2 Stakeholders

The stakeholders in this program are:

it@cork Skillnet offers a broad range of training courses to address the varied training needs of our member companies, who operate at all levels across the IT sector and beyond.¹

SoftTest is Ireland’s Independent Software Testing Special Interest Group. Its goal is to facilitate knowledge sharing within the Irish software testing community.

Program Members are the group of software testing professionals representing software companies based in the south of Ireland. At the time of writing, there are 23 members from industries as diverse as software services, healthcare, FinTech, security, gaming, computer hardware, insurance, biotech and HR systems.

1.3 Initiation

The first meeting in November had two goals:

1. To introduce the participants and outline the difficulties faced by the Program Members
2. To identify the skills and capabilities required to achieve a professional and productive level of competence

James Trindle of McAfee introduced the session with a brief presentation summarizing the problems currently faced by software teams in acquiring and retaining talented testers. His talk offered a stark prospect, and in fact he called it an existential crisis.

Paul Gerrard facilitated the discussion of these problems to define the scope of the challenges faced. The meeting then split into smaller groups to brainstorm the skills requirements for a professional tester.

¹*it@cork* is a leading not for profit independent organization representing the Tech Sector in the South of Ireland. *it@cork* manages *it@cork Skillnet*, which is funded by Skillnet Ireland and member company contributions. Skillnet Ireland is a national agency dedicated to the promotion and facilitation of workforce learning in Ireland. It is funded from the National Training Fund through the Department of Education and Skills.

2 Why a New Tester Skills Program?

This chapter starts with the existential crisis that companies face when hiring and retaining testers. Later sections provide a wider industry view and a proposed new skills set.

2.1 *Existential Crisis for Testers*

Testing Is Obsolete The general feeling was that the approaches offered by training providers, books and the certification scheme(s) are no longer fit for purpose. They are outdated and have not kept pace with the industry changes experienced by all members.

Replaced by Automation A common perception is that testers and testing in general can be replaced by automated approaches. Managers are swayed by the promise of Continuous Delivery (CD), pervasive automation and the emergence of machine learning and artificial intelligence. Testers have not found a way to articulate the reasons why testing is much harder to automate and eliminate than people believe.

How Do You Add Value to the Team? If you ask a tester what value they bring to their teams, they find it extremely difficult to make a strong case. The argument that testing is important is won already. But how do testers explain their value? My experience is that almost all testers don't know who their customers (stakeholders) are; they don't know what stakeholders want or how the information testers provide is used to make decisions still. As a profession (if testing actually is a profession) we have failed to make the case.

Titles Changing—Evolution of SDET Role Companies are implementing various ways of redistributing testing in their teams. The Shift-Left idea has caught on, with developers taking more responsibility, testers acting as coach/mentors to devs and other team members and being more closely involved in requirements. These are all good things. More popular in the US than Europe, the SDET (Software Development Engineer in Test) role is a hard one to fill. What is clear is that testing is in a state of flux. Testers are finding it hard to assimilate the change and to contribute towards it.

We're All Engineers; Everyone Must Write Code Related to the SDET approach, testers who never wrote code (and might not ever want to) are being encouraged to learn a programming or scripting language and automated test execution tools. The pressure to program and use tools is significant. This is partly because of the relentless marketing of the vendors. But it is also fuelled by a lack of understanding of what tools can and should do and what they cannot and should

not do. Testers (whether they use tools or not) are not well briefed in the case for automation or the strategies for successful tool implementation.

Once Highly Respected Skillset/Mindset no Longer Valued The Year 2000 threat caused many companies to take testing seriously and there was a brief period when testers were more highly respected. But when Agile appeared on the scene and was widely adopted, the role of testers was badly or not defined. They were expected to just ‘get on with it’. Agile, at the start at least, was mostly driven as a developer initiative with little thought for how requirements and testing were done. After 15 years, testers have a much better idea of their role in Agile. Unfortunately, the next big thing is Continuous Delivery. The mantra is that testing, of whatever type, should be automated. Once again, testers are under pressure to re-define their role and/or get out of the project.

Technology Changing at Unprecedented Rate There’s little doubt that test approaches have not kept pace with the changing technology. Although test execution tools appear within a year or two of new user interface technologies, the new risks, modelling approach and test management methods emerge very slowly. Tester skills seem to be tied to technologies. Skills should be independent of technology, enabling testers to test *anything*.

2.2 *The Drive to Digital*

Across the business world, there is a revolution in the way that IT is being specified, developed, implemented and used. There is lots of hype around the whole ‘Digital Transformation’ phenomena. Digital Transformation programs are affecting business across all industry and government sectors. There is no doubt that it also affects people in their daily lives.

Digital includes traditional IT but includes:

- Mobile anything
- The Internet of Things
- Autonomous vehicles
- Our home, workplace, public and private spaces
- Robots (physical)
- Bots (software)
- Artificial Intelligence, Machine Learning, Deep Learning
- And so on . . .

Digital visionaries promise a lot:

- No human intervention in your systems (Autonomous Business Models)
- Marketing messages created, sent, followed up and changed almost instantly
- Full range of data from the smallest locale to global in all media formats at your disposal

- Autonomous drones, trucks and cars can transport products, materials and people
- Physical products needn't be ordered, held in stock and delivered at all—3D printing removes constraints.

Mankind has built some pretty impressive systems (of systems). The most complex systems ever used to be the Space Shuttle with 2.5 m parts, but this was superseded by the Nimitz class supercarrier which has one billion parts. In fact, the carrier comprises thousands of interconnected systems and with a crew of 5000–6000, it could be compared to an average town—afloat.

Compare this with a 'Smart City'. A Smart City is

an urban development vision to integrate multiple information and communication technology (ICT) and IoT solutions in a secure fashion to manage a city's assets—the city's assets include, but are not limited to, local departments' information systems, schools, libraries, transportation systems, hospitals, power plants, water supply networks, waste management, law enforcement, and other community services. (Wikipedia)

With a smart city, the number of connected nodes and endpoints could range from a million to billions. The smart city will be bigger, more complex than anything before. Connected citizens and many of the systems:

- Move in the realm of the city and beyond
- Interact in unpredictable ways
- Are places where citizens are not hand-picked like the military; crooks, spies and terrorists can usually come and go as they please

Unlike ships, smart cities are highly vulnerable to attack.

Digital systems will have a social impact on all citizens who encounter them. There are huge consequences as systems become more integrated with the fabric of society. Systems already monitor our every move, our buying, browsing and social activities. Bots push suggestions of what to buy, where to shop, who to meet, when to pay bills to us minute by minute.

Law enforcement is affected too. CCTV monitors traffic, people and asset movement and our behaviours. The goal might be to prevent crime by identifying suspicious behaviour and controlling the movement of law enforcement agents to places of high risk. But these systems have the potential to infringe our civil liberties and the legal frameworks are behind the technology.

Digital affects all industries.

Unlike Agile, which is an ongoing IT initiative, Digital is driven by Business. Agile has taken more than 15 years to get halfway implemented in the software

industry. Digital has taken no time at all—perhaps 2–3 years and it is all-pervasive in the West. Digital is the buzz-phrase of the moment.

The speed of delivery is partly about pro-action, but it is also about survival.

Often, Chief Digital Officers are Marketers. Marketers move at the pace of marketing and they want change at the same pace. To the marketer, frequent software delivery is critical. Mobile users *expect* apps to change almost daily with new features, offers, opportunities appearing all the time. Users often don't care which supplier they use, as long as their apps work reliably, so businesses are in an APPS RACE.

S/W development at the pace of marketing.

So, automation (and not just test automation) is critical. What business needs is IT responsiveness—what you might call *true agility*. This doesn't necessarily mean hundreds of releases every day; but it does mean business want rapid, regular turnaround from ideas to software delivery.

With continuous integration/deployment, DevOps, developers can now *promise Continuous Delivery*

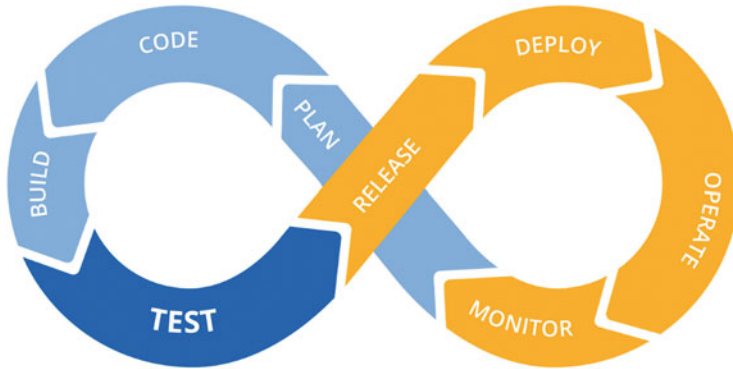
Testers need to provide *Continuous Assurance*

This means automation through the (shortened) life cycle. What exactly is possible and impossible with automation, right here, right now? Are Continuous Delivery and DevOps the route to success? Could testing be the bottleneck that prevents success? How do testers operate in dynamic, high-paced, automation-dominated environments?

2.3 Waterfall Thinking Won't Work with Continuous Methods

Continuous Delivery or an adapted version of it is becoming increasingly popular in Digital projects and if Digital is the 'future' for the majority of organizations, then we had better prepare for it. Testers need to adapt to fit into their continuous delivery regimes so let's look at how continuous approaches are normally described.

The most common diagram one sees is the figure eight or infinite loop below. The principle is that the plan, code, build, test through release, deploy, operate and monitor phases are sequential but are repeated for every release.



But there’s a problem here. If you unwrap the infinite loop, you can see that the phases are very much like the stages of a Waterfall development. There are no feedback loops, you have to assume one phase completes before another starts.



So, it appears that Continuous Delivery is just Waterfall in the small. What do we *know* about waterfall-style developments?

- *It’s sequential*—one stage follows another—no variation
- *Dependencies rule*—you can’t start one stage before the previous stage is done
- *It’s not re-entrant*—no flexibility to react to external events
- *Testing has stages itself*—we *know* that testing has itself stages of thinking and activities spread through the process
- *Only one phase of testing*—but there are developer and tester manuals and automated test execution activities
- *Testing is squeezed*—timeboxed activities—the thinking, preparation and execution time is all limited
- *No feedback loop(s)*—we know that testing finds bugs—but the continuous process has no feedback loop.

If Agile has taught us anything, it’s that the dependence on staged approaches made Waterfall unsuccessful in more dynamic environments.

Staged thinking won't work in a continuous process.

We need another way of looking at process to make Continuous Delivery work.

2.4 *Separating Thinking from Logistics*

There are two problems to solve here:

1. The first is that there is no one true way or best practice approach to implementing, for example, continuous delivery. Everyone does it slightly differently, so any generic training scheme has to offer generic practices.
2. The second is that any credible training scheme must recognize that there are skills that can be taught in the classroom, but the *employer* must take on the role of explaining local practices and embedding skills.

These local practices are what we call *logistics*. Logistics are how a principled approach is applied locally. Locally might mean 'across an entire organization' or it could mean every project you work on works differently. If you work on multiple projects, therefore, you will have to adapt to different practices—even if you are working in the same team.

Principles and thinking are global; logistics are local.

It's clear that to offer training alone is not enough. There must be a contribution by the local employer to nurture trainees, coach them in local practices and give them work that will embed the skills and local approach.

To offer training to practitioners, we must separate the principles and thinking processes from the logistics.

How do we do this?

2.5 *Logistics: We Don't Care*

We need to think clearly and remove logistics from our thinking. The simplest way to do this is to identify the aspects of the local environment and descope them, so to speak. The way Paul usually introduces this concept is to identify the things that we don't care about.

As a practitioner you will care deeply about logistics, but for the purposes of identifying the things that are universally applicable—principles and our thought process—we need to set them aside for the time being. Here are the key logistical aspects that we must remove 'to clear our minds'.

Document or Not? We don't care whether you document your tests or not. Whether and how you record your tests is not germane to the testing thought process.

Automated or Manual? We don't care whether you run tests by hand, so to speak, or use a tool, or use some form of magic. It isn't relevant to the thought process. The mechanism for test execution is a logistical choice.

Agile vs. Waterfall? We don't care whether you are working in an Agile team or in a staged, waterfall project or are part of a team doing continuous delivery. It's not relevant to the testing thought process.

This Business or That Business? We don't care what business you are in whether it is banking or insurance or healthcare or telecoms or retail. It doesn't matter.

This Technology vs. That Technology? We don't care what technology you are working with. It's just not relevant to the thought process.

Programmer or Tester? We don't care who you are—developer, tester, user business analyst—the principles of testing are universal.

Test Manager or No Test Manager? We don't care whether you are working alone or are part of a team, with or without a test manager overseeing the work. This is a logistical choice, not relevant to the testing thought process.

2.6 *Without Logistics: The New Model for Testing*

If we dismiss all these logistics—what's left? Some people might think we have abandoned everything, but we haven't. If you set aside logistics, what's left is what might be called the universal principles and the thought process. Now, you might think there are no universal principles. But there clearly are—they just aren't muddied by local practices. Paul's book, *The Tester's Pocketbook* [4, 5] identifies 16 so-called *Test Axioms*.

Some Axioms, for example the stakeholder axiom, 'Testing Needs Stakeholders', are so fundamental they really are self-evident. Other axioms such as the Sequencing axiom, 'Run our most valuable tests first—we may not have time to run them later',

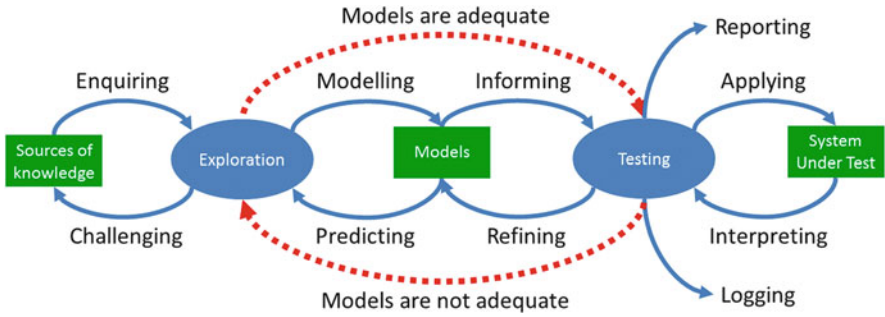


Fig. 1 The new model for testing

are more prosaic—it sounds logistical. But sequencing is a generally good thing to do—HOW you prioritize and sequence is your logistical choice.

The New Model for Testing is an attempt to identify the critical testing thought processes (Fig. 1). A Webinar [5] and white paper [6] give a full explanation of the thinking behind the model, which is reproduced below.

The model doesn't represent a process with activities, inputs, outputs, entry and exit criteria and procedures. Rather it represents the *modes of thinking* which people who test go through to achieve their goals. Our brains are such wonderful modelling engines that we can be thinking in multiple modes at the same time and process thoughts in parallel. It might not be comfortable, but from time to time, we must do this.

The New Model suggests that our thinking is dynamic and event-driven, not staged. It seems like it could be a good model for testing dynamic and event-driven approaches like continuous delivery.

Using the New Model as the basis for thinking fits our new world of testing.

The ten thinking activities all have associated real activities (logistics usually) to implement them and if we can improve the way we think about the testing problem, we are better able to make informed choices of how we logistically achieve our goals.

There are several consequences of using the New Model. One aspect is how we think about status. The other is all about skills.

2.7 Rethinking Status

As a collaborative team, all members of the team must have a shared understanding of the status of, for example, features in the pipeline. Now, the feature may be placed

Status is what we are thinking and doing, not where a feature is on a board

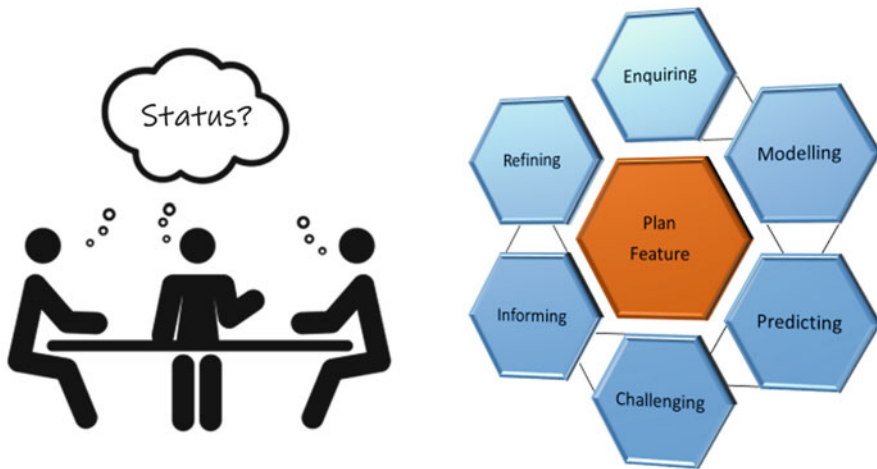


Fig. 2 Status is what we are thinking

somewhere on a Kanban or other board, but does the location on the board truly represent the status of the feature?

Consider the ‘three amigos’ of user, developer and tester. When it comes to agreeing status, it is possible that the user believes their job is done—the requirement is agreed. The developer might be writing code—it’s a work-in-progress. But the tester might say, ‘I have some outstanding challenges for the user to consider. There are some gaps in the story and some ambiguous scenarios we need to look at’ (Fig. 2).

What is the status of the feature? Done? Work in Progress? Or under suspicion? When all three participants are thinking the same way, then there is a consensus on the status of the feature.

Continuous Collaboration is an essential part of continuous delivery. The New Model provides a framework for meaningful discussion and consensus.

2.8 The (Current) Problem with Certification

The most prominent tester certification scheme is created and administered by the International Software Testing Qualifications Board—ISTQB [7]. The training classes run and qualifications awarded number 875,000+ and 641,000, respectively.

There are some minor schemes which operate, but ISTQB has become the de facto standard in the industry.

But there are well-known problems with current certification:

- If you look at the certification scheme syllabuses (Foundation and Advances Test Analyst, for example), the table of contents comprises mostly what we have called logistics. *The certification schemes teach many of the things we say we do not care about.*
- The schemes mostly offer one way of implementing testing—they are somewhat aligned with various standards. Incident Management, Reviews, Test Planning, Management and Control are all prescriptive and largely based on the Waterfall approach.
- Much of the syllabus and course content is about remembering definitions.
- The syllabuses infer test design techniques are procedures, where the tester never models, or makes choices. The tester and exam-taker are spoon fed the answers rather than being encouraged to think for themselves. There is no modelling content.
- The syllabuses don't teach *thinking skills*. The word *thinking* appears once in the 142 pages of Foundation and Advanced Test Analyst syllabuses.
- Exams, being multiple choice format, focus on remembering the syllabus content, rather than the competence of the tester.

Certification does not improve your ability to be independent thinkers, modelers or (pro-)active project participants; the exams do not assess your ability as a tester.

This is a big problem.

3 The Tester Skills Program

After that lengthy justification for a new approach to thinking and inevitably skills acquisition, this chapter focuses on the strategy for the development of a Tester Skills Program (TSP).

Work started on the TSP in late 2018 in line with a provisional plan agreed with it@cork Skillnet. But as time passed it became obvious that the initial goal of creating a 3-day beginner class would not satisfy the requirements of the Program Members. The challenges faced and the range of topics required to fulfil the needs of a professional tester were much more ambitious than could be delivered in just 3 days. With hindsight, this was obvious, but we tried to align with the plan as agreed.

The current strategy emerged over the early months.

3.1 *Skills Focus*

There were several influences on the depth and range of skills needed and consequently, there are a range of objectives for the program:

- The syllabus would focus on non-logistics, that is the principles, more independent thinking, modelling and people skills. (As a consequence, there is little overlap with ISTQB syllabuses.)
- Practitioners need skills that allow them to work in teams that use pervasive automation for environments, builds, tests.
- Practitioners might be expected to work in teams where continuous, event-driven processes are emerging or being adopted.
- The range of skills implies a broader role of assurance is required which spans requirements through to testing in production; new disciplines of shift-left, continuous delivery, Digital eXperience Optimization (DXO) require consideration.
- Practitioners are assumed to be part of mixed, multidisciplinary teams and must have basic project, collaboration, interpersonal and technical skills.

Skills should align with a better-defined goal of testing: to increase definition and delivery understanding.

3.2 *New Tester Behaviours*

Program members wanted the scheme to encourage specific behaviours of practitioners:

- To think more analytically (modelling, systems and critical thinking)
- To move from passive to active collaboration; to challenge and refine requirements
- To understand customer or digital experience optimization; to be aware of and exploit other predictive models and align testing to these models
- To act like a pathfinder or navigator (rather than a ‘follower’)
- To collaborate with confidence and at more senior technical and business/stakeholder levels

The TSP syllabus aims to encourage more outward-facing, collaborative, proactive behaviour.

3.3 TSP Is a Learning and Development Scheme

Clearly, TSP is more than a few days classroom training. The training must be part of a more comprehensive L&D regime. Training can impart new ideas, concepts and skills, but to trigger new behaviours, these skills must be embedded in the practitioners' mind and aligned with local ways of working.

For every hour of training material, there needs to be 1–2 h support, assignments and practical work to achieve the goal of new behaviours. Employers are encouraged to support learners by answering their questions and providing local logistics knowledge to complete the learning process.

When learners do receive line manager support, 94% go on to apply what they learned. There's a positive correlation between the transfer of learning to the workplace, line manager support and performance improvement.

(Kevin Lovell, Learning Strategy Director at KnowledgePool)

3.4 The Skills Inventory

The initial work of the Program Members was to define the range of skills required for a testing practitioner. It was understood at the outset that the range of skills meant that there had to be a graduated set of L&D schemes. The Skills Inventory would be a shopping list of topics that could be part of Foundation, Advanced or Mastery level schemes.

The summary Topic Areas in the inventory appear below:

- Adapting Testing
- Advanced Testing
- Agile Testing Approaches
- Assertiveness
- Certification
- Challenging Requirements
- Collaboration
- Communication
- Critical Thinking
- Developer Testing
- Exploratory Testing
- Exploring Sources of Knowledge
- Facilitation
- Hiring Testers
- Instrumentation
- Modelling
- Monitoring
- Non-Functional Testing
- Process Improvement

- Planning
- Reconciliation
- Regression Testing
- Requirements Test Analysis
- Risk Management
- SDET Role
- Technical Testing
- Technology Skills
- Systems Thinking
- Test Assurance
- Test Automation Frameworks
- Test Automation
- Coaching
- Test Design—Model-Based
- Test Design—Domain
- Test Design—State-Based
- Test Design—Logic
- Test Design—Purposeful Activity
- Test Motivation
- Test Strategy
- Testability
- Testing and Stakeholders
- Testing Fundamentals
- Testing in Teams
- Working Remotely

As you can see, there are quite a few topics that you won't find on common test training courses. Personal and professional development topics include Critical Thinking, Assertiveness, Collaboration, Communication, Facilitation, Hiring, Process Improvement, Systems Thinking, Coaching, Testing and Stakeholders, Testing in Teams and Working Remotely.

3.5 Program Member Challenges

As part of the discussion of the Existential Crisis, the Program Members identified a range of challenges that face them. Not everyone has the same challenges, but the list below gives an indication of the kind of problems being faced in tester recruitment, education and retention. For each challenge, the relevant skills topic area(s) have been assigned. Understanding these challenges is helping a lot to define the syllabus topics. In this way, the syllabus focuses on the right problems.

Challenges	Skills areas
Tester candidates: on paper look great, but they don't seem to be able to analyse a requirement and be a tester	Requirements Test Analysis, Testing Fundamentals
Instilling sense that you need to test your work, college kids want to write code, but not test their own work	Test Motivation, Testing Fundamentals
Agile teams, dev and test teams work as 'agile' but not together	Testing in Teams
Cafeteria agile, teams choose to do what teams like to do	Adapting Testing to Change, Agile Testing
Lots of hands on deck to get things delivered, but who leads team? Who leads on testing?	Test Leadership, Testing in Teams
Is the tester the lead on quality? If not, who is?	Testing in Teams
Everyone does their own thing, but who sets the strategy? Gaps and overlaps?	Testing in Teams, Test Strategy
Brief sprint—devs want to hand off asap, but testers are left behind, left with questions	Testing in Teams
Changed focus from testing to quality engineering—but what is quality engineering?	Adapting Testing
From tester in team responsible for automation and acceptance, should they move towards being a test coach, testmaster?	Coaching
Risk analysis, exploration	Risk Management
Skills needed: coaching, exploration and risk management	Coaching
Architecting for testability	Testability
TDD and role in test strategy	Developer Testing
Good design for test—what is it? How to recognize and encourage developers	Developer Testing, Testability
Exploratory testing, critical mindset, seeing the difference in confirmation and challenging the product	Exploratory Testing, Critical Thinking, Test Motivation
Coaching, within the team, devs and users, but also across teams, encouraging observability, logging, instrumentation	Coaching, Monitoring, Instrumentation
Macro-level consistency checks, instrumentation and logging	Monitoring, Instrumentation, Reconciliation
Influencing teams—helping them to spot flaws, educating, leading teams, facilitation	Coaching, Facilitation
Role for policing?	Test Assurance
Leading retrospectives, continuous improvement, leaving room for innovation and improvization	Coaching, Process Improvement
Balance between control and innovation	Process Improvement
Does coaching operate only on a local level?	Coaching
Is there a difference between coaching testers and developers? At an individual vs. team level?	Coaching
Challenging requirements, user stories, etc.	Challenging Requirements
Waterfall test has no voice. In agile we have a voice. To be pro-active is a matter of critical thinking but also guts	Testing in Teams, Assertiveness, Critical Thinking

(continued)

Coaching up as well as down (managers)	Coaching
Communication skills, how to articulate questions and information	Communication
Critical thinking, and influencing	Critical Thinking, Communication
Critical testing skills	Testing Fundamentals
Collaboration skills, testers create tests and automation in isolation. Testers bring in pairing, risk discussions, bug bashes	Collaboration, Communication
Testing as an activity not a role, but organizations exist to achieve outcomes	Testing and Stakeholders
‘There are people who lose their ability to think’, don’t ask questions, don’t challenge	Test Motivation
Critical thinking	Critical Thinking
Understanding what devs do well and testers/SDETSs do well (and not so well)	Developer Testing, SDET Role
High-performing teams have devs writing automation	Test Automation
What do stakeholders need from testers?	Testing and Stakeholders
Can you teach exploratory testing? Is it a mindset? Or an aptitude?	Exploring Sources of Knowledge, Exploratory Testing
BBST is a much more in-depth regime. Big investment of time.	Advanced Testing
Coach in all directions	Coaching
Devs need better test skills too!	Developer Testing
Mastery vs. capability?	Advanced Testing
How long to achieve (assessment of?) capability?	Certification, Advanced Testing
How long to achieve (assessment of?) mastery?	Certification, Advanced Testing
Qualification, certification?	Certification, Hiring Testers
A lot of testers lack the confidence/aptitude to explore and to question software	Test Motivation, Testing Fundamentals, Exploratory Testing
Most people are not critical thinkers	Test Motivation, Critical Thinking
Certification used as a tick-box to impress employers, not to improve skills	Certification
Testers unwilling to learn, improve	Test Motivation, Adapting Testing
Reluctance to test below the UI	Technical Testing, Testability
Unwillingness to test ‘outside the box’	Test Motivation, Modelling
Inability to demonstrate the value of testing	Testing and Stakeholders, Test Motivation

3.6 *Structure of the Foundation Scheme*

At the time of writing, only the Foundation Level scheme is defined, although the syllabus and course content are still work-in-progress. The high-level syllabus appears over the page.

The teaching content comprises approximately 40 h of material; this would be supported by 40–80 h of assignments and offline discussion with peers and managers, reading and further research. Much of the early training, on test fundamentals gives learners a set of questions to ask and discuss with peers, managers and stakeholders.

Beyond that, the assignments tend to be one of the following:

- Research, reading and study on testing-related issues
- Topics such as test design have specific assignments, such as requirements to analyse and online applications to explore and test.
- Modelling (focusing on requirements, stories, software, usage, tests)
- Practical test assignments (exploration, test design and bug-finding)

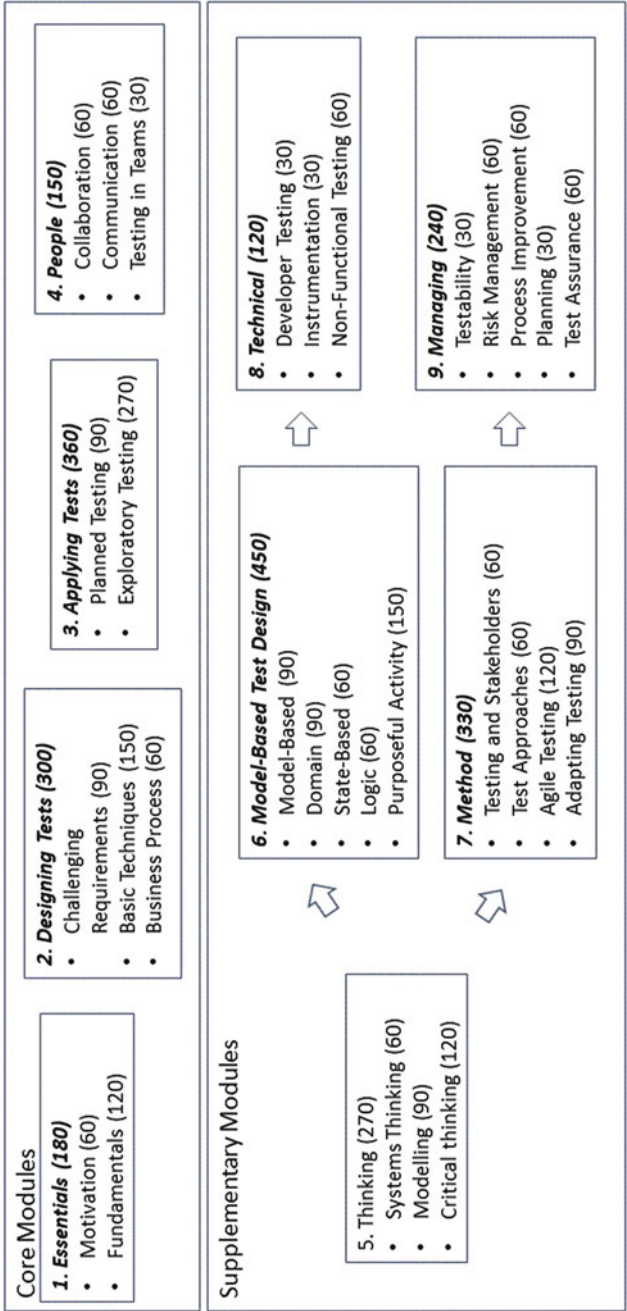
The scale of teaching is such that our expectation is that most companies would opt for a mix of classroom instructor-led, online instructor-led and purely online teaching. ALL training material will be presentable in all three formats. For the initial pilot classes, the ‘core’ modules would be presented in a classroom and feedback obtained. It is anticipated that the non-core modules would be usually accessed online.

Since a focus of the scheme is to help people to adapt to dynamic project environments, the thrust of the training is to help testers to think for themselves. A core component of this is the systems, critical and testing thinking modules. Not every learner will be comfortable with all of this material, and systems thinking in particular focuses on broader problem-solving than just testing. But exposure to systems thinking is still deemed to be of positive benefit.

Finally, there are some ‘people skills’ modules. These are intended to provide insights into the challenges of working in teams, collaboration and basic communication skills. At the Foundation level these are purely introductory and the intention is to get learners to at least pay attention to people issues. The Advanced scheme is likely to go into much more depth and offer specific personal skills modules.

Overall, the goal of the Foundation level is to bring new hires with little or no testing experience to a productive level. We have not used Bloom’s taxonomy to assign learning objectives, but the broad goal is for learners to achieve K4 level ‘Analysis’ capability in the analysis and criticism of requirements and the selection of models and modelling and testing of systems based on requirements or exploratory surveys. *Compare this with the ISTQB Foundation, where the learning objectives are split K1: 27%. K2: 59%, K3: 14%—a far less ambitious goal [8].*

Tester Skills Program – FOUNDATION
Version 0.4.12 June 2019



4 The Future

4.1 Engaging Competent Training Providers to Deliver TSP

One of the deliverables of the scheme is a template ‘Invitation to Tender’ for the Foundation level training. it@cork Skillnet require a means of inviting training providers to build and deliver classes against the TSP Syllabuses. In this way, companies can compete to deliver training, and this should ensure prices are competitive.

One of the shortcomings of the certification schemes is the sheer size of the syllabuses. The ISTQB Foundation and Advanced Test Analyst documents add up to 142 pages. The TSP Foundation syllabus—roughly equivalent in duration, if not content, will be less than 10% of that length.

One of the reasons the ISTQB syllabuses are so over-specified is that it allows non-testing professionals to deliver certified training. The thrust of the TSP is that the materials are bound to be delivered by experienced testing professionals (who may have left their testing career behind but are nevertheless qualified to deliver good training). The TSP Foundation allows trainers to teach what they know, rather than ‘what is written’ and means trainers are in a good position to answer the tricky questions that are built into many of the assignments.

4.2 Could the TSP Be the Basis of Certification?

This currently isn’t on the agenda in the Program Membership. However, the ISTQB scheme which relies entirely on multiple choice exams and requires no employer or peer review or practical experience could be improved upon.

TSP at least attempts to engage peers and managers in the support of learners. Certainly, the assignments could be compiled into a ‘workbook’ which could contain a summary of the assignments, verified by a peer, manager or mentor. These could demonstrate that learners have done their homework at least.

In common with professional engineering bodies, a formal certification scheme would require some proof of relevant work experience and perhaps some original work relating to the experience or further study in a subject related to the TSP topic areas. The mechanical, civil or electrical engineering professional bodies in the UK (or elsewhere) could be examined perhaps to derive a scheme that would go some way towards making testing a real profession.

4.3 Classroom, Online Instructor-Led or Online Teaching?

The goal to date is to offer all course materials in a format that could be used in a classroom or online delivery. Although most practitioners would prefer classroom courses, the flexibility and lower cost of online training is attractive. The choice is likely to be driven by whether the student is, for example, self-employed or employed by a larger organization preferring self-study or instructor-led teaching.

The market will determine what formats are most appropriate and commercially successful.

4.4 Proliferating the TSP

Could the TSP be adopted as worldwide standard? Time will tell. But there are opportunities to proliferate the TSP scheme [9].

Irish Commercial Adoption The Program Members intention is to adopt the Foundation scheme for their graduate intake. it@cork Skillnet will partially subsidize the training and encourage the scheme to be adopted across Ireland. It is to be hoped that local training providers create (or cross-license) training material and ‘train their trainers’ to deliver classes.

University/College Adoption It would be extremely helpful to have universities and colleges teach the TSP program as part of their curriculums. This would allow the academic institutions to make their graduates more marketable and remove some less attractive content. Although there are early discussions in Ireland, the general view is the academics won’t move until the TSP scheme is a proven success. Time will tell.

Overseas Commercial Adoption Gerrard consulting will promote the scheme in both Ireland and the UK in the belief that the scheme offers a significant uplift in the quality and value of testing and assurance related education. Being the facilitator and first provider of training gives a commercial advantage, but to scale up the scheme, other providers must come into the market.

Creating One’s Own TSP It goes without saying, that the process of development of a local TSP could be repeated in other regions. There is no reason, in principle, why other regions can’t identify their own local skills needs and build their own program. The ‘not invented here’ syndrome may also have an effect and bigger training providers might want exclusive rights to their own scheme in their locality.

It would be nice to think that a global TSP standard could be the ultimate outcome of this work, but commercial pressures and human nature suggest different schemes might spring up if the Irish TSP is deemed to be a success.

References

1. Skillnet Ireland. <https://www.skillnetireland.ie>. Accessed 19 July 2019
2. it@cork. <https://www.itcork.ie>. Accessed 19 July 2019
3. <https://softtest.ie>. Accessed 19 July 2019
4. <https://testaxioms.com>. Accessed 19 July 2019
5. Agile Testing Days Webinar. The new model for testing. <https://youtu.be/1Ra1192OpqY>. Accessed 19 July 2019
6. A New Model for Testing – white paper. <https://gerrardconsulting.com/mainsite/wp-content/uploads/2019/06/NewModelTestingIntro.pdf>. Accessed 19 July 2019
7. Gerrard, P.: The Tester’s Pocketbook. Testers’ Press, Maidenhead, UK (2011)
8. <https://istqb.org>. Accessed 19 July 2019
9. <https://testerskills.com>. Accessed 19 July 2019

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.

