Certified Tester

Agile Technical Tester

Overview of Syllabus

Version 1.1

International Software Testing Qualifications Board

Revision History

Version	Date	Remarks
Syllabus v0.1	11 Jan 2017	Standalone sections
Syllabus v0.2		WG review comments on v01 incorporated
Syllabus v0.3		WG review comments on v02 incorporated
Syllabus v0.7		Alpha review comments on v03 incorporated
Syllabus v0.71		Working group updates on v07
Syllabus v0.9		Beta version
Syllabus 2017		GA version
Syllabus 2019	10 Jul 2019	Changes
Syllabus 2019	14 Sep 2019	Cosmetic changes
Syllabus 2019	14 Nov 2019	Release Date
Syllabus 2020	28 Jan 2020	Adapt Header and remove unnecessary text

Table of Contents

Acknowledgements

This document was produced by a team from the International Software Testing Qualifications Board Agile Working Group lead by Rex Black (chair), Michaël Pilaeten (Vice-chair and chair a.i.) and Renzo Cerquozzi (Product Owner).

The Advanced Level Agile team thanks the review team and the National Boards for their suggestions and input.

At the time the Advanced Level Agile Technical Tester syllabus was completed, the Working Group had the following membership: Michaël Pilaeten (Chair a.i.), Renzo Cerquozzi (Product Owner), Alon Linetzki (Marketing workgroup), Leanne Howard (Glossary workgroup) and Klaus Skafte (Exam workgroup).

Authors: Leo van der Aalst, Renzo Cerquozzi, Bertrand Cornanguer, István Forgács, Jani Haukinen, Noam Kfir, Sammy Kolluru, Alon Linetzki, Tilo Linz, Michaël Pilaeten, Marie Walsh.

Internal Reviewers: Michael Arefi, Vojtěch Barta, Renzo Cerquozzi. Graham Bath, Laurent Bouhier, Anders Claesson, Alessandro Collino, David Janota, David Evans, Leanne Howard, Matthias Hamburg, Kari Kakkonen, Tor Kjetil Moseng, Meile Posthuma, Salvatore Reale, Marko Rytkönen, Sarah Savoy, Klaus Skafte, Mike Smith, Chris van Bael.

Creation and Review of Sample Questions: Armin Born, Renzo Cerquozzi, Alon Linetzki, Tilo Linz, Jamie Mitchell, Jani Haukinen, Tobias Horn, Chris van Bael, Suruchi Varshney,

The team thanks also the following persons, from the National Boards and the Agile expert community, who participated in reviewing, commenting, and balloting of the Agile Extension Syllabus: Adam Roman, Armin Beer, Beata Karpinska, Chris Van Bael, Erwin Engelsma, Giancarlo Tomasig, Gary Mogyorodi, Ingvar Nordström, Jana Gierloff, Jörn Münzel, Jurian van de Laar, Kari Kakkonen, Laurent Bouhier, Marko Rytkönen, Martin Klonk, Matthias Hamburg, Meile Posthuma, Paul Weymouth, R. Green, Richard Seidl, Rik Marselis, Stephanie Ulrich, Stephanie van Dijck, Tal Pe'er, Tilo Linz, Veronica Seghieri and Wim Decoutere.

A special thanks to Galit Zucker, ISTQB® general secretary, for the Guidance and Support.

This document was formally approved for release by the General Assembly of the ISTQB® on October 18th, 2019

0. Introduction to this Syllabus

0.1 Purpose of this Document

This syllabus forms the basis for the International Software Testing Qualification at the Advanced Level for the Agile Technical Tester. The ISTQB® provides this syllabus as follows:

- To National Boards, to translate into their local language and to accredit training providers. National Boards may adapt the syllabus to their particular language needs and modify the references to adapt to their local publications.
- To Exam Boards, to derive examination questions in their local language adapted to the learning objectives for each syllabus.
- To training providers, to produce courseware and determine appropriate teaching methods.
- To certification candidates, to prepare for the exam (as part of a training course or independently).
- To the international software and systems engineering community, to advance the profession of software and systems testing, and as a basis for books and articles.

The ISTQB® may allow other entities to use this syllabus for other purposes, provided they seek and obtain prior written permission.

0.2 Overview

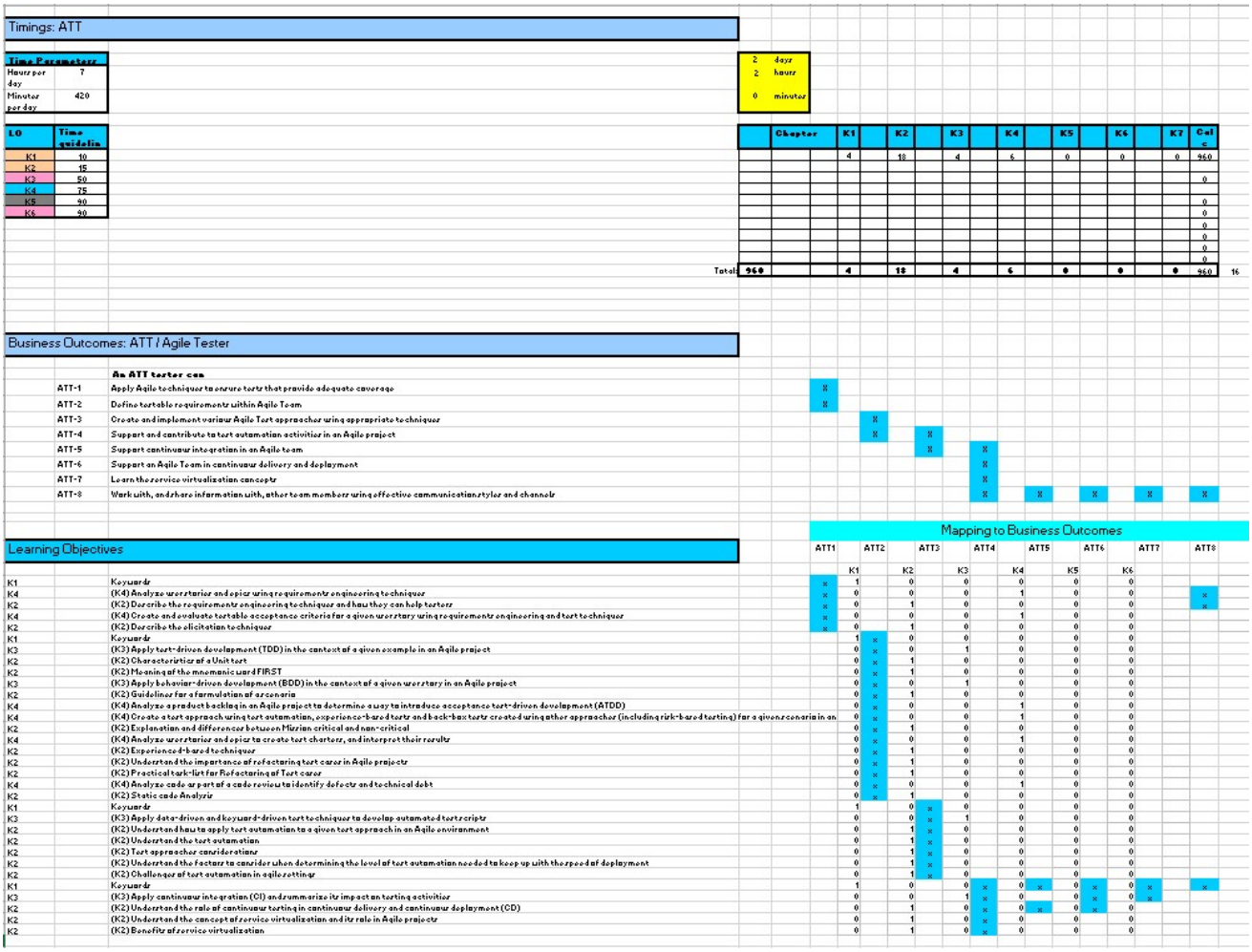
The Advanced Level Agile Technical Tester Overview document (this document) includes the following information:

- Business Outcomes for the syllabus
- Summary for the syllabus
- Relationships among the syllabi
- Description of cognitive levels (K-levels)
- Appendices

0.3 Examinable Learning Objectives

The Learning Objectives support the Business Outcomes and are used to create the examination for achieving the Certified Tester Advanced Level—Agile Tester Technical Certification. In general, all parts of this syllabus are examinable at a K1 level. That is, the candidate will recognize, remember, and recall a term or concept. The specific learning objectives at K1, K2, K3, and K4 levels are shown at the beginning of the pertinent chapter.

1. Timing Spreadsheet



Timings: ATT

Time Parameters	
Hours per day	7
Minutes per day	420

2	days
2	hours
0	minutes

LO	Time guideline
K1	10
K2	15
K3	50
K4	75
K5	90
K6	90

	Chapter	K1		K2		K3		K4		K5		K6		K7	Calc
		4		18		4		6		0		0		0	960
															0
															0
															0
															0
															0
															0
Total:	960	4		18		4		6		0		0		0	960

16

Business Outcomes: ATT / Agile Tester

	An ATT tester can
ATT-1	Apply Agile techniques to ensure tests that provide adequate coverage
ATT-2	Define testable requirements within Agile Team
ATT-3	Create and implement various Agile Test approaches using appropriate techniques
ATT-4	Support and contribute to test automation activities in an Agile project
ATT-5	Support continuous integration in an Agile team
ATT-6	Support an Agile Team in continuous delivery and deployment
ATT-7	Learn the service virtualization concepts
ATT-8	Work with, and share information with, other team members using effective communication styles and channels

Learning Objectives

Mapping to Business Outcomes

		ATT1	ATT2	ATT3	ATT4	ATT5	ATT6	ATT7	ATT8
		K1	K2	K3	K4	K5	K6		
K1	Keyword	1	0	0	0	0	0		
K4	(K4) Analyze user stories and epics using requirements engineering techniques	x	0	0	0	1	0		x
K2	(K2) Describe the requirements engineering techniques and how they can help testers	x	0	1	0	0	0		x
K4	(K4) Create and evaluate testable acceptance criteria for a given user story using requirements engineering and test techniques	x	0	0	0	1	0		
K2	(K2) Describe the elicitation techniques	x	0	1	0	0	0		
K1	Keyword		1	0	0	0	0		
K3	(K3) Apply test-driven development (TDD) in the context of a given example in an Agile project		x	0	1	0	0		
K2	(K2) Characteristics of a Unit test		x	1	0	0	0		
K2	(K2) Meaning of the mnemonic word FIRST		x	1	0	0	0		
K3	(K3) Apply behavior-driven development (BDD) in the context of a given user story in an Agile project		x	0	1	0	0		
K2	(K2) Guidelines for formulation of scenarios		x	1	0	0	0		
K4	(K4) Analyze a product backlog in an Agile project to determine a way to introduce acceptance test-driven development (ATDD)		x	0	0	1	0		
K4	(K4) Create a test approach using test automation, experience-based tests and black-box tests created using other approaches (including risk-based testing) for a given scenario in an		x	0	0	1	0		
K2	(K2) Explanation and difference between Mizrian critical and non-critical		x	1	0	0	0		
K4	(K4) Analyze user stories and epics to create test charters, and interpret their results		x	0	0	1	0		
K2	(K2) Experienced-based techniques		x	1	0	0	0		
K2	(K2) Understand the importance of refactoring test cases in Agile projects		x	1	0	0	0		
K2	(K2) Practical task-list for Refactoring of Test cases		x	1	0	0	0		
K4	(K4) Analyze code as part of a code review to identify defects and technical debt		x	0	0	1	0		
K2	(K2) Static code Analysis		x	1	0	0	0		
K1	Keyword	1		0	x	0	0		
K3	(K3) Apply data-driven and keyword-driven test technique to develop automated test scripts	0		0	x	1	0		
K2	(K2) Understand how to apply test automation to a given test approach in an Agile environment	0		1	x	0	0		
K2	(K2) Understand the test automation	0		1	x	0	0		
K2	(K2) Test approaches considerations	0		1	x	0	0		
K2	(K2) Understand the factors to consider when determining the level of test automation needed to keep up with the speed of deployment	0		1	x	0	0		
K2	(K2) Challenges of test automation in agile setting	0		1	x	0	0		
K1	Keyword	1		0	0	x	0	x	x
K3	(K3) Apply continuous integration (CI) and summarize its impact on testing activities	0		0	1	x	0	x	
K2	(K2) Understand the role of continuous testing in continuous delivery and continuous deployment (CD)	0		1	0	x	0	x	
K2	(K2) Understand the concept of service virtualization and its role in Agile projects	0		1	0	x	0		
K2	(K2) Benefits of service virtualization	0		1	0	x	0		

2. Advanced Agile Technical Tester – 960 mins.

Keywords

- Chapter 1: Acceptance criteria, epic, user story
- Chapter 2: Test-driven development, behavior-driven development, acceptance test-driven development, specification by example, test charter
- Chapter 3: data-driven testing, keyword-driven testing, test procedure, test approach
- Chapter 4: Service virtualization, continuous testing

2.1 The Advanced Agile Technical Tester LO's

ATT-1.x	(K1) Keywords
ATT-1.1.1-1	(K4) Analyze user stories and epics using requirements engineering techniques
ATT-1.1.1-2	(K2) Describe the requirements engineering techniques and how they can help testers
ATT-1.1.2-1	(K4) Create and evaluate testable acceptance criteria for a given user story using requirements engineering and test techniques
ATT-1.1.2-2	(K2) Describe the elicitation techniques
ATT-2.x	(K1) Keywords
ATT-2.1.1-1	(K3) Apply test-driven development (TDD) in the context of a given example in an Agile project
ATT-2.1.1-2	(K2) Understand the characteristics of a Unit test
ATT-2.1.1-3	(K2) Understand the meaning of the mnemonic word FIRST
ATT-2.1.2-1	(K3) Apply behavior-driven development (BDD) in the context of a given user story in an Agile project
ATT-2.1.2-2	(K2) Understand how to manage guidelines for a formulation of a scenario
ATT-2.1.3	(K4) Analyze a product backlog in an Agile project to determine a way to introduce acceptance test-driven development (ATDD)
ATT-2.2.1-1	(K4) Analyze the creation of a test approach using test automation, experience-based tests and back-box tests created using other approaches (including risk-based testing) for a given scenario in an Agile project
ATT-2.2.1-2	(K2) Explain differences between Mission critical and non-critical
ATT-2.2.2-1	(K4) Analyze user stories and epics to create test charters, and interpret their results
ATT-2.2.2-2	(K2) Understand the use Experienced-based techniques
ATT-2.3.1-1	(K2) Understand the importance of refactoring test cases in Agile projects
ATT-2.3.1-2	(K2) Understand practical task-list for Refactoring Test cases
ATT-2.3.2-1	(K4) Analyze code as part of a code review to identify defects and technical debt
ATT-2.3.2-2	(K2) Understand Static code Analysis
ATT-3.x	(K1) Keywords
ATT-3.1.1	(K3) Apply data-driven and keyword-driven test techniques to develop automated test scripts
ATT-3.1.2	(K2) Understand how to apply test automation to a given test approach in an Agile environment
ATT-3.1.3-1	(K2) Understand the test automation
ATT-3.1.3-2	(K2) Understand differences between various test approaches
ATT-3.2.1-1	(K2) Understand the factors to consider when determining the level of test automation needed to keep up with the speed of deployment
ATT-3.2.1-2	(K2) Understand the challenges of test automation in agile settings
ATT-4.x	(K1) Keywords
ATT-4.1.1	(K3) Apply continuous integration (CI) and summarize its impact on testing activities
ATT-4.1.2	(K2) Understand the role of continuous testing in continuous delivery and continuous deployment (CD)
ATT-4.2.1-1	(K2) Understand the concept of service virtualization and its role in Agile projects
ATT-4.2.1-2	(K2) Understand the benefits of service virtualization

Please refer the ATT syllabus for the up-to-dated list of LOs and explanation.

3. References

3.1 Standards

- [DO-178B] RTCA/FAA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, 1992.
- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE), 2005.

3.2 ISTQB® Documents

- [ISTQB_ALTA_SYL] ISTQB® Advanced Level Test Analyst Syllabus, Version 2012
- [ISTQB_ALTM_SYL] ISTQB® Advanced Level Test Manager Syllabus, Version 2012
- [ISTQB_FA_OVIEW] ISTQB® Foundation Level Agile Tester Overview, Version 1.0
- [ISTQB_FL_SYL] ISTQB® Foundation Level Syllabus, Version 2011

3.3 Books

- [Adzic09] Adzic, Gojko. “Bridging the Communication Gap” Neuri Ltd, 2009
- [Adzic11] Adzic, Gojko. “Specification by Example” Manning, 2011.
- [Beck02] Kent Beck, “Test Driven Development: By Example”, 2002
- [Beck99] Beck, Kent. “Extreme Programming Explained” Addison Wesley, 1999.
- [Carkenord08] Barbara A. Carkenord, “Seven Steps to Mastering Business Analysis”, J. Ross Publishing, 2008.
- [Cohn 09] Mike Cohn: Succeeding with Agile: Software Development Using Scrum. Addison-Wesley Professional, 2009;
- [Crispin08] Crispin, L. and Gregory, J. (2008) Agile Testing: A Practical Guide for Testers and Agile Teams. Crawfordsville: Addison-Wesley Professional
- [Elfriede99] Dustin Elfriede, Automated Software Testing: Introduction, Management, and Performance: Introduction, Management, and Performance, Addison-Wesley Professional, 1999
- [Evans03] Eric Evans, “Domain-Driven Design: Tackling Complexity in the Heart of Software”, 2003
- [Fewster12] Mark Fewster, Dorothy Graham, Experiences of Test Automation: Case Studies of Software Test Automation, Addison-Wesley Professional, 2012
- [Fowler/Parsons 10], Martin Fowler, Rebecca Parsons, Domain-Specific Languages, Addison-Wesley Signature, Series, 2010
- [Hendrickson13] Elisabeth Hendrickson, “Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing”, Pragmatic Bookshelf, 2013
- [Jeffries00] Ron Jeffries, Ann Anderson, and Chet Hendrickson, “Extreme Programming Installed,” Addison-Wesley Professional, 2000
- [Jorgensen13] Paul C. Jorgensen, Software Testing: A Craftsman’s Approach, Auerbach Publications; 4th edition, 2013
- [Linz14] Tilo Linz, Testing in Scrum, A Guide for Software Quality Assurance in the Agile World, Rocky Nook, 2014
- [Meszaros07] Gerard Meszaros, “xUnit Test Patterns: Refactoring Test Code”, Addison-Wesley, 1st Edition, Apress, 2007

- [Michelsen12] John Michelsen and Jason English, “Service Virtualization: Reality is Overrated”, Apress, 1st Edition, 2012.
- [Osherove09] Roy Osherove, “The Art of Unit Testing”, 2009
- [Paskal15] Greg Paskal, “Test Automation in the Real World: Practical Lessons for Automated Testing”, MissionWares, 2015
- [Smart15]; Smart, John Ferguson; "BDD in action", Manning, 2015
- [Wein89] Weinberg, Gerald & Gause, Donald. “Exploring Requirements: Quality Before Design” Dorset House, 1989.
- [Whittaker09] James A Whittaker, “Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design”, Addison-Wesley Professional, 2009
- [Wiegers02] Karl Wiegers, “Peer Reviews in Software: A Practical Guide (Paperback)”, 2002.

3.4 Agile Terminology

Keywords which are found in the ISTQB® Glossary are identified at the beginning of each chapter. For common Agile terms, we have relied on the following well-accepted Internet resources which provide definitions.

http://guide.Agilealliance.org/
http://whatis.techtargetcom/glossary
http://www.scrumalliance.org/

We encourage readers to check these sites if they find unfamiliar Agile-related terms in this document. These links were active at the time of release of this document.

3.5 Other References

The following references point to information available on the Internet and elsewhere. Even though these references were checked at the time of publication of this syllabus, the ISTQB® cannot be held responsible if the references are not available anymore.

- [Agile Alliance Guide] Various contributors, http://guide.Agilealliance.org/.
- [Agilemanifesto] Various contributors, www.agilemanifesto.org
- [Hendrickson]: Elisabeth Hendrickson, “Acceptance Test-driven Development,” testobsessed.com/2008/12/acceptance-test-driven-development-atdd-an-overview.
- [INVEST] Bill Wake, “INVEST in Good Stories, and SMART Tasks,” xp123.com/articles/invest-in-good-stories-and-smart-tasks.
- [Kubaczkowski] Greg Kubaczkowski and Rex Black, “Mission Made Possible,” www.rbcs-us.com/images/documents/Mission-Made-Possible.pdf.
- [Cohn09] The Forgotten Layer of the Test Automation Pyramid, https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid
- [CyclomaticComplexity] https://en.wikipedia.org/wiki/Cyclomatic_complexity
- [Farley] http://www.davefarley.net/?cat=5
- [DSL] https://en.wikipedia.org/wiki/Domain-specific_language
- [Fowler07] https://martinfowler.com/articles/mocksArentStubs.html
- [Fowler04] Fowler, Martin. https://martinfowler.com/bliki/SpecificationByExample.html
- [Fowler03] Martin Fowler, https://martinfowler.com/bliki/TechnicalDebt.html
- [Gherkin] https://docs.cucumber.io/gherkin/
- [IQBBA] International Qualifications Board for Business Analysts, http://www.iqbba.org/
- [IREB] International Requirements Engineering Board, https://www.ireb.org/en
- [IIBA] International Institute of Business Analysts (IIBA), https://www.iiba.org/
- [Marick01] Marick, Brian, http://www.exampler.com/old-blog/2003/08/21/#agile-testing-project-1
- [Marick03] Marick, Brian. http://www.exampler.com/old-blog/2003/08/22.1.html
- [North06] Dan North, “Introducing BDD”, blog post, 2006
- [TESTDOUBLES] Wojciech Bulaty, Bill Wake, “Stubbing, Mocking and Service Virtualization Differences for Test and Development Teams”, https://www.infoq.com/articles/stubbing-mocking-service-virtualization-differences
- [ToolList] Test tool review, information platform on the international market of software testing tools, www.testtoolreview.de/en/
- [xUnit] https://en.wikipedia.org/wiki/XUnit, https://en.wikipedia.org/wiki/Unit_testing