

Certified Tester Specialist Syllabus

Foundation Level

Acceptance Testing

Version 2019

International Software Testing Qualifications Board



Provided by International Qualification Board for Business Analysis



Copyright Notice

Copyright Notice © International Software Testing Qualifications Board (hereinafter called ISTQB®)
ISTQB® is a registered trademark of the International Software Testing Qualifications Board.

Copyright © 2019 the authors: Bruno Legeard (chair), Olivier Denoo, Debbie Friedenber, Anne Kramer, Karolina Zmitrowicz

All rights reserved. The authors hereby transfer the copyright to the ISTQB®. The authors (as current copyright holders) and ISTQB® (as the future copyright holder) have agreed to the following conditions of use:

Extracts, for non-commercial use, from this document may be copied if the source is acknowledged. Any Accredited Training Provider may use this syllabus as the basis for a training course if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus and provided that any advertisement of such a training course may mention the syllabus only after official Accreditation of the training materials has been received from an ISTQB®-recognized Member Board.

Any individual or group of individuals may use this syllabus as the basis for articles and books, if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus.

Any other use of this syllabus is prohibited without first obtaining the approval in writing of the ISTQB®.

Any ISTQB®-recognized Member Board may translate this syllabus provided they reproduce the abovementioned Copyright Notice in the translated version of the syllabus.

Revision History

Version	Date	Remarks
Beta	August 24 th , 2018	Candidate beta version
ISTQB® GA 2019	March 18 th , 2019	Candidate general release version after Beta review comments incorporated
2019	June 21 st , 2019	Release version
2019	May 21 st 2019	Change to copyright

Table of Contents

Copyright Notice	2
Revision History.....	3
Table of Contents	4
Acknowledgements	6
0 Introduction to this Syllabus.....	7
0.1 Purpose of the Document.....	7
0.2 Focus of the Syllabus	7
0.3 Business Outcomes.....	8
0.4 Examinable Learning Objectives	8
0.5 Recommended Training Times.....	9
0.6 Handling of Standards	9
0.7 Entry Requirements.....	9
0.8 Sources of Information.....	9
1 Introduction and Foundations – 80 mins.....	10
1.1 Fundamental Relationships	10
1.1.1 Business Goals, Business Needs and Requirements.....	10
1.1.2 Requirements / User Stories, Acceptance Criteria and Acceptance Tests	11
1.1.3 The Importance of the Quality of the Requirements	11
1.2 Business Analysis and Acceptance Testing	12
1.2.1 Relationship between Business Analysis and Testing Activities.....	12
1.2.2 Collaboration between Business Analysts and Testers in Acceptance Testing	13
1.2.3 How Acceptance Testing Can Drive the Development Process: ATDD and BDD	14
2 Acceptance Criteria, Acceptance Tests and Experience- Based Practices – 165 mins.	15
2.1 Writing Acceptance Criteria	15
2.2 Designing Acceptance Tests	16
2.2.1 Test Techniques for Acceptance Testing	16
2.2.2 Using the Gherkin Language to Write Test Cases	16
2.3 Experience-based Approaches for Acceptance Testing	17
2.3.1 Exploratory Testing.....	17
2.3.2 Beta Testing	18
3 Business Process and Business Rules Modeling – 150 mins.....	19
3.1 Modeling Business Processes and Rules.....	19
3.2 Deriving Acceptance Tests from Business Process/Rule Models.....	20
3.3 Business Process Modeling for Acceptance Testing	20
3.3.1 Good Practices for Business Process Modeling for Acceptance Testing	21
3.3.2 Using Business Process Models for ATDD	21
4 Acceptance Testing for Non-Functional Requirements – 95 mins.	23
4.1 Non-functional Characteristics and Quality in Use.....	23
4.1.1 Non-functional Quality Characteristics and Sub-characteristics	24

4.1.2 Quality in Use	25
4.2 Usability and User Experience.....	26
4.2.1 UX Requirements Analysis	26
4.2.2 Usability Testing	27
4.3 Performance Efficiency.....	27
4.3.1 High-level Performance Acceptance Tests.....	27
4.3.2 Acceptance Criteria for Performance Acceptance Tests	28
4.4 Security.....	28
5 Collaborative Acceptance Testing – 110 mins.....	30
5.1 Collaboration	30
5.2 Activities	31
5.2.1 Defect Analysis.....	31
5.2.2 Reporting.....	31
5.2.3 QA Activities for Acceptance Testing.....	32
5.3 Tool Support.....	32
6 Abbreviations.....	34
7 Registered Trademarks	35
8 References	36
Standards	36
IQBBA® Documents.....	36
ISTQB® Documents	36
Referenced in this Syllabus	37
Appendix A – Subset of BPMN 2.0.1 and DMN 1.2.....	38
A.1 Subset of BPMN 2.0.1	38
A.2 Subset of DMN 1.2	39

Acknowledgements

This document was produced by a core team consisting of volunteers from both the IQBBA® and ISTQB® organizations.

The core team thanks the following persons from the IQBBA® and ISTQB® member boards who participated in reviewing, commenting, and balloting for this syllabus: Chris Van Bael, Graham Bath, Renzo Cerquozzi, Ernst von Düring, Florian Fieber, Karol Frühauf, Beata Karpinska, Ine Lutterman, Elke Mai, Rik Marselis, Judy McKay, Jörn Münzel, Petr Neugebauer, Ingvar Nordström, Monika S. Olsen, Tal Pe'er, Lara Pellegrino, G. Pistarini, Meile Posthuma, Miroslav Renda, Jan Sabak, Lucjan Stapp, Richard M. Taylor, Stephanie Ulrich, Robert Werkhoven, Paul Weymouth.

This document was formally approved for release by the ISTQB® on 3 May 2019. This document is also approved by the IQBBA® scheme for the certification of Business Analysts' qualifications.

0 Introduction to this Syllabus

0.1 Purpose of the Document

This syllabus forms the basis for the ISTQB® Foundation Level Acceptance Testing syllabus certification.

The ISTQB® provide this syllabus as follows:

1. To member boards, to translate into their local language and to accredit training providers. Member boards may adapt the syllabus to their particular language needs and modify the references to adapt to their local publications.
2. To exam boards, to derive examination questions in their local language based on the learning objectives.
3. To training providers, to produce training materials and determine appropriate teaching methods.
4. To certification candidates, as a source to prepare for the exam.
5. 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 Focus of the Syllabus

Assessment and validation of the business solution are important and frequent activities of product owners (POs), business analysts (BAs) and testers. Part of their responsibilities is to define acceptance criteria for the requirements, regardless of the type of development lifecycle – Agile or traditional. Acceptance criteria are defined by decomposing the requirements into a more atomic and testable form. Test cases are then designed to verify the solution against the criteria. Designing acceptance tests from acceptance criteria should be a highly collaborative activity, involving business analysts and testers, to ensure high business value of the acceptance testing phase, and mitigating the risks related to product release.

Supporting this collaborative work, and thereby avoiding the silo effect between product owners / business analysts and testers, is the key objective of this syllabus.

This acceptance testing qualification is aimed at anyone involved in software acceptance testing activities. This includes people in roles such as product owners, business analysts, testers, test analysts, test engineers, test consultants, test managers, user acceptance testers, and software developers.

The focus of the syllabus is on the concepts, methods and practices of collaboration between product owners / business analysts and testers in acceptance testing. Regarding the different forms of acceptance testing defined in the ISTQB® Certified Tester

Foundation Level syllabus, this syllabus covers user acceptance testing (UAT), contractual and regulatory acceptance testing as well as alpha and beta testing. This syllabus voluntarily does not address the operational acceptance testing (OAT) because this is generally performed by teams that will operate the system rather than by testers and business analysts.

0.3 Business Outcomes

This section lists the Business Outcomes expected of a candidate who has achieved the Foundation Level Acceptance Testing Certification.

As this qualification covers three roles, business analysts, product owners, and testers, the business skills targeted are expressed by role.

For business analysts and product owners:

- AcT-1 Contribute to an organization's acceptance testing activities by participating in the acceptance test design phase and supporting the alignment of the product with the business requirements.
- AcT-2 Contribute to the organization of acceptance testing activities, including the process, artifacts, communication, reporting, monitoring, and management of such activities, and collaborate with testers and other relevant stakeholders in this process.
- AcT-3 Contribute to the quality of the acceptance testing process, including validation and verification of produced artifacts.

For testers:

- AcT-4 Contribute to the definition of acceptance criteria during the requirements definition phase.
- AcT-5 Collaborate efficiently with business analysts and other stakeholders during all acceptance testing activities.
- AcT-6 Understand the business objectives, communicate with business units, and share common objectives for acceptance testing.

0.4 Examinable Learning Objectives

The Learning Objectives support the Business Outcomes and are used to create examinations for achieving the ISTQB® Foundation Level Acceptance Testing Certification. Learning objectives are allocated to a cognitive level of knowledge (K-Level).

A K-level, or Cognitive level, is used to classify learning objectives according to the revised taxonomy from Bloom [Anderson01]. ISTQB® uses this taxonomy to design its syllabi examinations.

This syllabus considers three different K-levels (K1 to K3):

- K1 – Remember. The candidate should remember or recognize a term or a concept.
- K2 – Understand. The candidate should select an explanation for a statement related to the question topic.
- K3 – Apply. The candidate should select the correct application of a concept or technique and apply it to a given context.

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 learning objectives at K2 and K3 levels are shown at the beginning of the pertinent chapter.

0.5 Recommended Training Times

A minimum training time has been defined for each learning objective in this syllabus. The total time for each chapter is indicated in the chapter heading.

Training providers should note that other ISTQB® syllabi apply a “standard time” approach which allocates fixed times according to the K-Level. The Acceptance Testing syllabus does not strictly apply this scheme. As a result, training providers are given a more flexible and realistic indication of minimum training times.

0.6 Handling of Standards

Standards (IEEE, ISO, etc.) are referenced in this syllabus. The purpose of these references is to provide a source of additional information if desired by the reader. Please note that only the items from these standards that are referenced specifically in the syllabus are eligible for examination. The standards documents themselves are not intended for examination and are included only for reference.

Please refer to Chapter 8 for a list of referenced standards.

0.7 Entry Requirements

The ISTQB® Foundation Level certificate shall be obtained before taking the Foundation Level Acceptance Testing certification exam.

0.8 Sources of Information

Terms used in the syllabus are defined in ISTQB®’s Glossary of Terms used in Software Testing [ISTQB_GLOSSARY]. A version of the Glossary is available from the ISTQB®.

1 Introduction and Foundations – 80 mins.

Keywords

acceptance criteria, acceptance testing, requirement, user story

Learning Objectives

1.1 Fundamental Relationships

- AcT-1.1.1 (K1) Recall the relationship between business goals, business needs and requirements
- AcT-1.1.2 (K2) Explain the relationship between requirements / user stories, acceptance criteria and acceptance tests
- AcT-1.1.3 (K2) Explain how the quality of requirements / user stories and acceptance criteria affects acceptance testing

1.2 Business Analysis and Acceptance Testing

- AcT-1.2.1 (K2) Summarize the relationship between acceptance testing activities and business analysis activities
- AcT-1.2.2 (K2) Explain how testers and business analysts collaborate in acceptance testing activities
- AcT-1.2.3 (K2) Describe Acceptance Test-Driven Development (ATDD) and Behavior-Driven Development (BDD)

1.1 Fundamental Relationships

While it is certainly true that the roles and responsibilities of the tester and the business analyst are different, it is also true that their activities are complementary; work done by one group may greatly affect, either positively or negatively, that of the other. This is especially true in acceptance testing which is performed to assess the system's readiness for deployment and its use by the customer (end-user). Good collaboration between business analysts and testers is particularly important for a proper consideration of the business implications at this test level.

1.1.1 Business Goals, Business Needs and Requirements

Business analysts first must understand the organization's overall business goals and identify current business processes and stakeholders. Once that is done, they describe specific business needs and determine a business case that addresses those needs. Once this high-level work has been completed, requirements can be elicited for the business solution that shall be developed.

Business goals, business needs, business requirements, and product requirements (see [IQBBA® Glossary] for a definition of these four terms) describe, at different levels of

abstraction, what shall be achieved. In Agile development, the same principles apply, but different terms may be used (for example features and user stories).

In this document, the term “requirements” refers both to business requirements and to product requirements.

1.1.2 Requirements / User Stories, Acceptance Criteria and Acceptance Tests

During requirements elicitation, business analysts and testers (possibly together with developers) should begin to create specific acceptance criteria and develop acceptance tests as a joint effort. This ensures that there is a mutual understanding of what “acceptable” means from the business, development, and testing perspectives, right from the beginning of the project.

Acceptance criteria relate directly to a specific requirement or user story. They are either part of the detailed description or an attribute of the related requirement. If user stories are used, acceptance criteria are part of the user story’s definition and extend the story [ISTQB_FL_AT_SYL].

In all cases, acceptance criteria are measurable criteria, formulated as a statement (or a set of statements), which can be either true or false. They are used to check whether a requirement or user story has been implemented as expected. Acceptance criteria represent the test conditions which determine “what” to test. They do not contain the detailed test procedures.

Acceptance test cases are derived from acceptance criteria. These tests specify how the verification of the acceptance criteria should be performed.

1.1.3 The Importance of the Quality of the Requirements

If acceptance criteria and tests are based on requirements, user stories, and/or acceptance criteria that are vague or ambiguous, it is likely that testers will make assumptions about stakeholder expectations and business needs. In this case, the resulting acceptance tests may be flawed. This will lead to rework or, even worse, the running of invalid tests, thus creating unnecessary costs as well as risks and uncertainty about product quality assurance.

It is critical for testers to work closely with business analysts to make sure that requirements are clear and well understood by all stakeholders concerned. Ambiguities should be resolved and assumptions should be clarified so that the resulting acceptance tests are valid and are a meaningful way to determine the product’s readiness for release.

In Agile development, the INVEST criteria [Cohn04] define a set of criteria, or checklist, to assess the quality of a user story. These may be used by business analysts / product owners, developers, and testers to ensure the quality of user stories (cf. ISTQB® Foundation Level Agile Tester syllabus [ISTQB_FL_AT_SYL]). The ISO/IEC/IEEE

29148:2011 standard [ISO/IEC 29148:2011] provides good practices of the requirements engineering process to ensure the development of good quality requirements.

1.2 Business Analysis and Acceptance Testing

Too often, business analysts and testers work in their own separate silos, which can lead to misunderstandings about business and customer expectations. Those misunderstandings may stay hidden until the release approaches. By taking advantage of the complementary skills and by working together, business analysts and testers can positively affect the development process. This can be accomplished both by considering acceptance criteria and acceptance testing as early as possible and by coordinating efforts to make sure that the product has been tested appropriately prior to release at acceptance test level.

1.2.1 Relationship between Business Analysis and Testing Activities

The following are the main elements of the IQBBA® business analysis activities [IQBBA_FL_SYL]:

- Strategy definition
- Management of the business analysis processes
- Requirements engineering in business analysis
- Solution evaluation and optimization

The business analyst is responsible for identifying business needs of stakeholders and for determining solutions to business problems with the aim of introducing change which adds value to the business [IQBBA_FL_SYL]. An important aspect of the business analyst's role is to establish consensus between quality engineers, testers, developers, system integrators, product managers and project managers.

A test process consists of the following main groups of activities [ISTQB_FL_SYL]:

- Test planning
- Test monitoring and control
- Test analysis
- Test design
- Test implementation
- Test execution
- Test completion

Quite a few of the associated activities and tasks relate to both business analysis and testing. The following examples illustrate the relationship between the two disciplines in the context of acceptance testing:

Requirements engineering in business analysis vs. test planning, test analysis and test design:

- During the requirements engineering activities in business analysis, business analysts prepare detailed business and product requirements. These requirements are part of the test basis for the test planning, test analysis and test design activities, as testers define their objectives and plan their work, evaluate the specifications and requirements, identify test conditions and design test cases and test procedures.
- Testers can contribute to the definition and verification of acceptance criteria as part of test analysis and test design activities. Working together, the two roles ascertain that there is proper understanding of the solution and agree on the appropriate approach to acceptance testing.
- When requirements change, business analysts and testers can work together to assess the impact of the changes.

Solution evaluation in business analysis vs. test implementation, test execution and test completion:

- During the solution evaluation phase in business analysis, business analysts support test implementation and test execution activities. They review the testers' procedures/scripts, clarify issues and potentially help with creation of test data to support business-related tests.
- Business analysts can assist with the implementation and execution of the acceptance tests. They may also support testers by evaluating test results. In addition, they may assist testers in test completion activities.

There is a strong and symbiotic relationship between the two roles and their respective activities, starting at the very beginning of a project and continuing until acceptance or release of the solution.

1.2.2 Collaboration between Business Analysts and Testers in Acceptance Testing

The common goal for business analysts and testers is to support the production of products with the highest possible value for the customer. Given their position within the organization, business analysts and testers have various opportunities to collaborate during the acceptance testing activities described in the previous section. Apart from joint discussions and reviews of generated artifacts, business analysts and testers collaborate in other areas. For example, collaboration on test planning based on risk analysis is a good opportunity to ensure that the appropriate test cases will be developed and prioritized.

In addition to the direct benefits of working together and supporting each other's efforts during acceptance testing, there is an important opportunity to cross-train team members. The more testers know about business needs and stakeholder requirements, and the more business analysts know about structured testing, the more likely the two groups will understand and appreciate each other's work and better collaborate within the project.

1.2.3 How Acceptance Testing Can Drive the Development Process: ATDD and BDD

The wide acceptance of Agile software development practices has influenced how acceptance testing relates to requirements elicitation and other business analysis activities. In sequential lifecycle models, acceptance test analysis, design, and implementation are activities to be handled by the testers after the requirements are finalized. With the Agile lifecycle model, acceptance criteria and acceptance test cases are created during requirements analysis, requirements refinement sessions, and product backlog refinement. This allows the implementation of the "Early Testing" principle by using the design of test cases as part of the requirements definition activities.

In the following two approaches, acceptance test analysis and design are formally part of the requirements engineering process:

- In Acceptance Test-Driven Development (ATDD) [Pugh11], acceptance tests are produced collaboratively during requirements analysis by business analysts, product owners, testers and developers.
- Behavior-Driven Development (BDD) [Smart14] uses a domain-specific scripting language, Gherkin, that is based on natural language statements. The requirements are defined in a 'Given – When –Then' format. These requirements become the acceptance test cases and also serve as the basis for test automation. See Section 2.2.2 for more information on Gherkin.

Both of these approaches engage the entire Agile team and help to focus the development efforts on the business goals. The approaches also treat the acceptance test cases as living documentation of the product because they can be read and understood by business analysts and other stakeholders. Acceptance test cases represent scenarios of usage of the product.

The two approaches are similar and the two terms are sometimes used interchangeably. In practice, BDD is associated with the use of Gherkin to support writing acceptance tests, while ATDD relies on different forms of textual or graphic acceptance test design. For example, the graphical representation of application workflows may be used to implement a visual ATDD approach.

2 Acceptance Criteria, Acceptance Tests and Experience-Based Practices – 165 mins.

Keywords

beta testing, experience-based test technique, exploratory testing, keyword-driven testing, test case

Learning Objectives

2.1 Writing Acceptance Criteria

AcT-2.1.1 (K3) For a given requirement or user story, develop a set of acceptance criteria that meet good practices

2.2 Designing Acceptance Tests

AcT-2.2.1 (K2) Explain test approaches and test techniques for acceptance testing

AcT-2.2.2 (K3) Apply the Gherkin language for designing acceptance tests for a given user story

2.3 Experience-based Approaches for Acceptance Testing

AcT-2.3.1 (K2) Summarize how exploratory testing can be used for acceptance testing

AcT-2.3.2 (K2) Summarize the relationship between beta testing and acceptance testing

2.1 Writing Acceptance Criteria

Specifying acceptance criteria is an important acceptance testing task. It helps to refine requirements or user stories and provides the basis for acceptance tests. Business analysts and testers should collaborate closely on the specification of these criteria. This collaboration ensures high business value from the acceptance testing phase and increases the chance of a successful iteration or product release.

Writing acceptance criteria forces business analysts and testers to think about functionality, performance, and other characteristics from a stakeholder or usage perspective. This supports early verification and validation of the related requirement or user story and provides a better chance of detecting inconsistencies, contradictions, missing information or other problems.

The following good practices should be considered when writing acceptance criteria [Cohn04]:

- Well-written acceptance criteria are precise, measurable and concise. Each criterion must be written in a way that enables the tester to measure whether or not the test object complies with the acceptance criterion.
- Well-written acceptance criteria do not include technical solution details. They concentrate on the question "What shall be achieved?" rather than on the question "How shall it be achieved?".
- Acceptance criteria should address non-functional requirements (quality characteristics) as well as functional requirements.

As with requirements and user stories, acceptance criteria should be reviewed through walkthroughs, technical reviews, iteration planning meetings or other methods (if necessary).

2.2 Designing Acceptance Tests

This section addresses the test techniques and approaches frequently used for acceptance testing.

2.2.1 Test Techniques for Acceptance Testing

In a requirements-based approach to acceptance testing, the tester derives test cases from the acceptance criteria related to each requirement or user story using black-box techniques such as equivalence partitioning or boundary value analysis (see [ISTQB_FL_SYL] Chapter 4).

Acceptance testing may be augmented with other test techniques or approaches:

- Business process-based testing, possibly combined with decision table testing, validates business processes and rules (see Section 3.2).
- Experience-based testing leverages the tester's experience, knowledge and intuition (see Section 2.3.1)
- Risk-based testing is based on risk types and levels. Prioritization and thoroughness of testing depends on previously identified product risks.
- Model-based testing uses graphical (or textual) models to obtain acceptance tests [ISTQB_MBT_SYL].

Acceptance criteria should be verified by acceptance tests and traceability between the requirements / user story and related test cases should be managed.

2.2.2 Using the Gherkin Language to Write Test Cases

In ATDD and BDD, acceptance tests are often formulated in a structured language, referred to as the Gherkin language [Smart14]. Using the Gherkin language, test cases are phrased declaratively using a standardized pattern:

- Given [a situation]
- When [an action on the system]

- Then [the expected result]

The pattern allows business analysts, testers and developers to write test cases in a way that is easily shared with stakeholders and may be translated into automated tests.

The “Given” block aims to put the test object in a state before performing test actions in the “When” block. The “Then” block specifies the consequences that can be observed from the actions defined in the “When” block. Test cases written in Gherkin do not refer to user interface elements but rather to user actions on the system. They are structured natural language test cases that can be understood by all relevant stakeholders.

In addition, the structure “Given – When – Then” can be parsed in an automated way. This allows automated test script creation using a keyword-driven testing approach [ISTQB_FL_SYL].

Initially, Gherkin was specific to some software tools supporting BDD, but it is now synonymous with the “Given – When – Then” acceptance test design pattern.

2.3 Experience-based Approaches for Acceptance Testing

All experience-based test techniques described in [ISTQB_FL_SYL] are relevant for acceptance testing. This section is focused on how exploratory testing can be used for acceptance tests, and on beta testing as a source of feedback on system usage.

2.3.1 Exploratory Testing

Exploratory testing [Whittaker09] is an experience-based test technique that is not based on detailed predefined test procedures. In exploratory testing, all activities are carried out within an uninterrupted period of time called a session. The testers are domain experts. They are familiar with user needs, requirements and business processes, but they are not necessarily familiar with the product under test.

During an exploratory testing session, the tester accomplishes the following:

- Learns how to work with the product
- Designs the tests
- Performs the tests
- Interprets the results

It is a good practice in exploratory testing to use a test charter. The test charter is prepared prior to the testing session (possibly jointly by the business analyst and the tester) and is used by the person in charge of the exploratory session (either a business analyst, tester or another stakeholder). It includes information about the purpose, target, and scope of the exploratory session, the test setup, the duration of the session, and possibly some tactics to be used during the session (such as the type of user that shall be simulated during the exploratory session). Time-boxed sessions help to control the time and effort

dedicated to the exploratory session. It is also good practice to perform exploratory testing in pairs or as team work.

In Agile development, exploratory test sessions can be conducted during an iteration by the product owner and/or the testers for acceptance testing of user stories assigned to the iteration.

Exploratory testing should be used to complement other more formal techniques in acceptance testing. For example, it may be used to provide rapid feedback on new features before methodical testing is applied.

2.3.2 Beta Testing

Beta testing is a form of acceptance testing that is often used for Commercial Off-the-Shelf Software (COTS) or for Software as a Service (SaaS) platforms. It is conducted to obtain feedback from the market after development and inhouse testing are completed.

Unlike other acceptance testing forms, beta testing is performed by potential or existing users at their own location. Beta tests neither impose predefined test procedures nor a test charter. Apart from the observed findings, the test activities are usually not documented at all.

Because the product is tested in various realistic configurations by actual users in their business process context, beta testing may discover defects that escaped during the development process and previous test levels. Resolving issues found by beta tests helps organizations avoid costly hotfixes or product recalls on a larger scale.

Acceptance testing should not be limited to beta testing. Beta testing is not systematic or measurable. There is no guarantee that all requirements or user stories are covered by the tests. Moreover, beta testing is performed late in the development process whereas tests based on acceptance criteria support the “Early Testing” principle.

3 Business Process and Business Rules Modeling – 150 mins.

Keywords

coverage, model-based testing

Learning Objectives

3.1 Modeling Business Processes and Rules

AcT-3.1.1 (K3) Construct a simple business process/rule model* using BPMN and/or DMN notations

3.2 Deriving Acceptance Tests from Business Process/Rule Models

AcT-3.2.1 (K3) Derive a set of acceptance tests covering a given coverage criterion from a given, simple business process/rule model* (in BPMN or DMN)

3.3 Business Process Modeling for Acceptance Testing

AcT-3.3.1 (K2) Summarize the good practices of business process and business rule modeling for acceptance testing

AcT-3.3.2 (K2) Explain how business process and business rule modeling can be used for ATDD

* Note: "Simple business process/rule model" means a model with less than 20 modeling elements, using only the element types defined in the appendix of this syllabus.

3.1 Modeling Business Processes and Rules

Organizations need confidence that critical business processes, such as order-to-cash procedures, human resource on-boarding, or production planning, can be performed without disruption. This is known as "business process assurance" and it is an essential objective of acceptance testing. In this context, two standards exist that provide a common language for business analysts and testers for graphically representing business processes and business rules: Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN). These models support the design and implementation of tests and help to determine the priority for execution.

Business process/rule models describe the business flow and the expected behavior of the test object. Representing business processes and rules to be tested using a graphical notation helps to establish a common understanding of what is expected. A business process corresponds to a flow of tasks, alternative paths, and the various events at the start, the end or possibly during the control flow. Business rules define explicit criteria for guiding behavior, shaping judgments, or making decisions.

BPMN, maintained by the Object Management Group (OMG), is a recognized standard for business process modeling which uses a flowcharting technique [OMG BPMN 2.0] [ISO/IEC 19510:2013]. In this syllabus, a subset of the BPMN 2.0.1 notation is used that is sufficient to draw simple business process models in the context of acceptance testing activities. This subset is defined in Appendix A1.

DMN, also standardized by the OMG, is complementary to the BPMN standard [OMG DMN 1.2]. While BPMN is used to represent workflows, DMN is used to represent decisions, business rules and outcomes/output within the workflow. In this syllabus, a subset of the DMN 1.2 notation is used that is sufficient to define business rules in conjunction with simple business process models in BPMN 2.0. This subset is defined in Appendix A2.

3.2 Deriving Acceptance Tests from Business Process/Rule Models

A business process model with business rules, described with the BPMN 2.0 and/or DMN 1.2 notations, provides a precise definition of the scenarios to be tested, including the cases related to business rules. It is a good basis for generating acceptance tests using coverage-based test selection criteria as defined in a model-based testing approach [ISTQB_MBT_SYL].

Coverage-based test selection follows the principle that the business analyst and tester agree on the coverage items that shall be fully tested. Typical coverage items for business process models when generating acceptance tests include the following:

- User stories, requirements, and risks annotated in the business process model
- Decisions in the decision tables describing the business rules
- User scenarios defined by different paths through the business process model
- All paths (usually without loops) through the business process model

Once the coverage items are defined, the tester then identifies a set of test cases that covers those items. Full coverage is achieved if the test suite covers each occurrence of the coverage item in the model at least once during execution.

Different coverage criteria may be combined to meet the acceptance testing objectives. For example, the objective may be to cover all paths of a given main scenario, but only one path of each alternative scenario.

3.3 Business Process Modeling for Acceptance Testing

Business process/rule models describe the business flow and the expected behavior of the test object. The use of business process/rule modeling in the context of acceptance testing is based on good modelling practices and supports visual ATDD practices.

3.3.1 Good Practices for Business Process Modeling for Acceptance Testing

The following good practices should be considered when using BPMN and DMN notations for acceptance testing:

- It is not necessary to describe everything in a business process model. The graphical representations of business processes in BPMN should focus on requirements to be tested. Therefore, workflow descriptions that only partially cover the behavior of related software systems are acceptable, as long as they represent what is to be tested.
- Especially for rule-based business processes, using decision tables helps manage dependencies. DMN supports the definition of conditions and outcomes corresponding to the business rules under test.
- Diagrams should be as simple as possible and be structured in sub-processes when needed to limit the number of graphical elements in a single business process diagram. This improves readability and facilitates reviews.
- Business process modeling for acceptance testing should be a collaborative work between business analysts and testers. Artifacts produced should be shared between and reviewed by both roles. Early and close communication between those two roles improves the quality of requirements or user stories as well as tests. (This is true for all test levels.)
- Additional information such as links to user stories, requirements, risks, priorities and any other information useful for acceptance testing should be added to the diagrams using annotations. By keeping all relevant information in a single location, it becomes easier to make decisions and reasons are better documented.

3.3.2 Using Business Process Models for ATDD

During the refinement sessions for requirements and user stories, the business process and business rule models will help the team to get into the details of the expected behavior and the acceptance criteria. The representation of workflows in BPMN and of rules in DMN directly enable testers to design appropriate test cases that verify the acceptance criteria.

Business process modeling for ATDD is based on the following principles:

- Business analysts and testers collaborate to model workflows and business rules using graphical notations such as BPMN and DMN.
- These business process/rule models are reviewed with relevant stakeholders and contribute to the validation of the requirements and acceptance criteria.
- Testers derive tests from these business process/rule models to ensure and demonstrate the required coverage through the different paths and business rules.
- Business analysts and testers may also use the models to identify changes that necessitate test case maintenance and to select regression test cases.
- Business process/rule models created and maintained for ATDD can be viewed as living documentation used by business analysts to present the actual behavior of the test object.

- Automated test generation techniques can be used to produce and maintain automated test scripts [ISTQB_MBT_SYL]. The model-based testing approach can also be combined with keyword-driven testing and data-driven testing approaches [ISTQB_FL_SYL].

Business process/rule modeling in ATDD provides a visualization of the workflows to be tested. This is the major difference from the Gherkin language used in BDD (see Section 2).

4 Acceptance Testing for Non-Functional Requirements – 95 mins.

Keywords

performance efficiency, security, usability, user experience

Learning Objectives

4.1 Non-functional Characteristics and Quality in Use

- AcT-4.1.1 (K2) Give examples of ISO 25010 non-functional sub-characteristics that should be considered in acceptance tests
- AcT-4.1.2 (K1) Recall quality in use characteristics according to the ISO 25010 standard

4.2 Usability and User Experience

- AcT-4.2.1 (K2) Relate different types of usage scenarios to the four pillars of UX requirements analysis
- AcT-4.2.2 (K2) Summarize different methods for testing usability within the respective domains of application

4.3 Performance Efficiency

- AcT-4.3.1 (K2) Explain high-level performance tests according to given efficiency requirements
- AcT-4.3.2 (K2) Recognize the impact of different perspectives on performance acceptance criteria

4.4 Security

- AcT-4.4.1 (K2) Explain why security acceptance criteria and related high-level security tests are required for a project in accordance with a given security requirement

4.1 Non-functional Characteristics and Quality in Use

Acceptance testing should cover both functional and non-functional requirements. Non-functional requirements are becoming more important with respect to acceptance testing due to the increased use of software in everyday life, data-driven processes, and the development of integrated services that rely heavily on complex software systems and on systems of systems.

4.1.1 Non-functional Quality Characteristics and Sub-characteristics

Meeting the expectations for non-functional quality characteristics strongly influences user acceptance of a proposed solution. Even when the criticality depends on the context, not properly addressing these quality characteristics can result in severe issues such as customer dissatisfaction, lost sales, rejection of the solution, liability risks, and public exposure of the organization.

The ISO 25010 standard [ISO 25010:2011] introduces a system and software product quality model that categorizes product quality properties. It includes seven non-functional characteristics, which are further detailed in sub-characteristics. The following table lists the sub-characteristics associated with the non-functional characteristics in ISO 25010.

Non-functional Characteristic	Sub-characteristics
Performance efficiency	<ul style="list-style-type: none">• Time-behavior• Resource utilization• Capacity
Compatibility	<ul style="list-style-type: none">• Co-existence• Interoperability
Usability	<ul style="list-style-type: none">• Appropriateness recognizability• Learnability• Operability• User error protection• User interface aesthetics• Accessibility
Reliability	<ul style="list-style-type: none">• Maturity• Availability• Fault tolerance• Recoverability
Security	<ul style="list-style-type: none">• Confidentiality• Integrity• Non-repudiation• Accountability• Authenticity
Maintainability	<ul style="list-style-type: none">• Modularity• Reusability• Analyzability• Modifiability• Testability

Portability	<ul style="list-style-type: none">• Adaptability• Installability• Replaceability
-------------	--

All non-functional quality characteristics should be considered in acceptance testing. An acceptance testing strategy should define the selection and priorities for the non-functional characteristics and sub-characteristics to be tested in a given context.

In the following sections of this chapter, usability, performance, and security are described in more detail as specific approaches are sometimes required to obtain a desired level of coverage.

4.1.2 Quality in Use

The standard ISO 25010 defines also the quality in use model with five characteristics related to outcomes of an interaction with a system: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage.

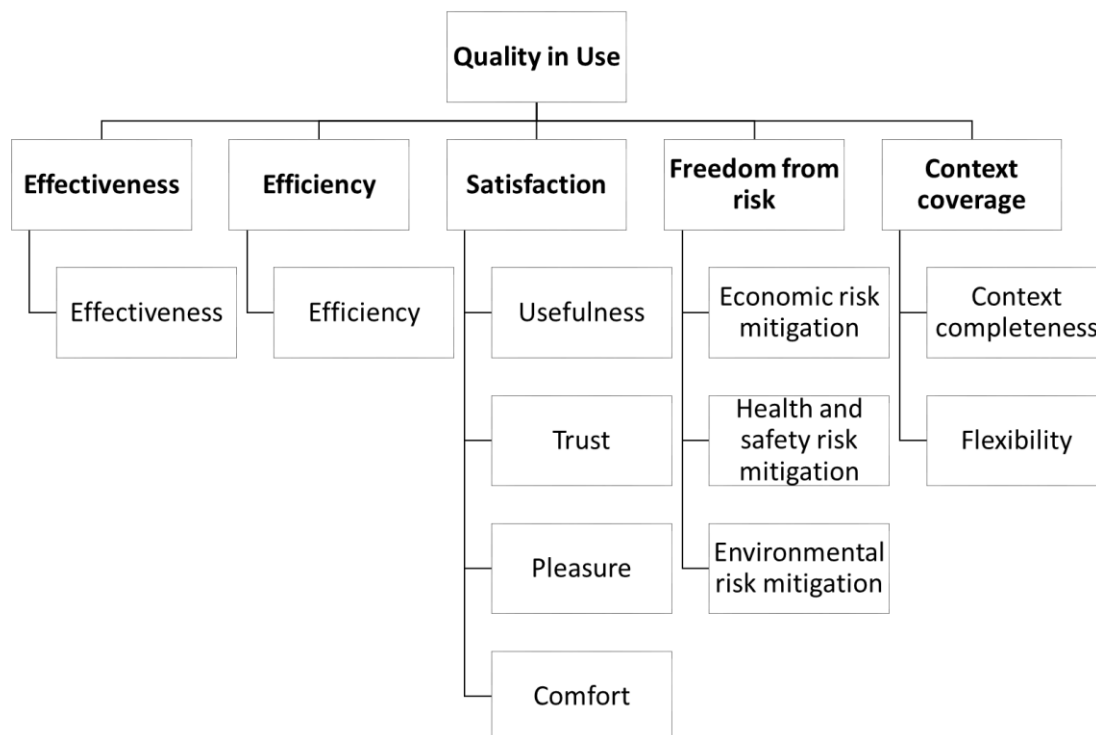


Figure 1 - Quality in Use model (extracted from [ISO 25010:2011])

Quality in use characteristics are particularly helpful in acceptance testing as they relate to the user's experience with the system.

4.2 Usability and User Experience

According to ISO 25010, usability is *“the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Usability can either be specified or measured as a product quality characteristic in terms of its sub-characteristics, or specified or measured directly by measures that are a subset of quality in use”*. [ISO 25010:2011].

Usability can be assessed against objectives (e.g., learnability, efficiency, memorability, satisfaction, error prevention) mostly in the form of heuristics [Rubin08] [Nielsen94]. The lack of usability may result in frustration, refusal to use the software and, in the most critical instances, in injury or death of the user.

User eXperience (UX) expands the term usability to include aesthetic and emotional factors such as an appealing, desirable design, aspects of confidence building, or satisfaction to use (e.g., pleasure, comfort). The context of using the system has a strong influence on the user experience as it may totally differ based on a number of factors such as location (e.g., the user is sitting behind a desk, driving a car or hiking), weather (e.g., sun, rain, cold), health conditions of the user (e.g., fatigue, age), environment (e.g., stressful, noisy).

Further details regarding usability testing are provided in the ISTQB Foundation Level Usability Testing syllabus [ISTQB_UT_SYL].

4.2.1 UX Requirements Analysis

UX requirements analysis is based upon the following four pillars:

- User analysis: Users are categorized in terms such as physical and intellectual characteristics, technical skills, business knowledge, socio-economic, and cultural background. Business analysts can also use models (e.g., based on personas [ISTQB_FL_AT_SYL]).
- Task analysis: Functionality is identified and formalized (e.g., through use cases and scenarios). User behavior and expectations are analyzed to design an optimized system or product.
- Context analysis: The context in which the system or product will be used is analyzed. External conditions (e.g. light, temperature, movement, humidity or dust), physical conditions (e.g., sitting, standing, lying, moving, hands-free) or “psychological” conditions (e.g. stress level, motivation, or the difference between private and professional usage) are considered to give directions to the subsequent design steps. Devices, platforms and form-factors (device-specific display) are also considered as part of the context.
- Competition analysis: Unless creating a disruptive design is the goal, business analysts should analyze the competitors and take inspiration from the successful implementation of their solutions to retain or attract users and customers. Another source of inspiration can come from successful solutions found in similar or even different sectors.

Due to common human limitations and biases (e.g., cognitive or perceptive biases, visual impairment, inexperience) [ISTQB_UT_SYL] some users might face more specific and sometimes severe difficulties in using software or products that are part of the business solution. Business analysts and testers should assess if products or services are accessible to all users by considering these limitations when designing acceptance criteria and test cases.

4.2.2 Usability Testing

There are different approaches to testing usability in acceptance testing:

- Checklist-based evaluations: Users evaluate the system or product under test according to checklists [Rubin08] to evaluate, compare and qualify their experience.
- Expert reviews: Usability experts evaluate the usability of the system or product according to pre-defined criteria or checklists based upon usability heuristics to identify strong and weak points of an interface.
- Walkthrough and think-aloud techniques: Users explore the product or systems and describe their actions and impressions out loud while doing so [ISTQB_UT_SYL]. They may be given specific tasks to accomplish to identify how they interact with the product and to learn about expectations or difficulties.
- Biometrics-based evaluations: User behavior is monitored with specific biometric devices (e.g., eye-movement recording, mouse-eye-movement recording) to understand how the user interacts with a page or a system, what attracts their attention, or what is more or less visible.
- Log files analysis: Retrospective analysis is conducted to review how the users interacted with the system to discover areas for possible improvement and to verify if actual use correlates with the intended profile/use.

4.3 Performance Efficiency

Performance efficiency (or simply “performance”) is an essential part of providing a “good experience” for users when they use their applications on a variety of fixed and mobile platforms [ISTQB_PT_SYL]. Performance tests must be considered at all levels of testing. During acceptance testing, performance tests are particularly addressed during Operational Acceptance Testing (OAT), usually by the operating teams. However, business analysts and testers should also be involved when developing acceptance criteria and related test cases. Acceptance criteria for performance efficiency requirements should provide objective measures, thus avoiding subjective performance evaluation during acceptance test execution.

4.3.1 High-level Performance Acceptance Tests

Performance testing aims to determine a system’s responsiveness and stability under certain conditions. In a typical performance test, concurrent users or transactions are simulated with specific tools to generate a given workload which mimics, as closely as possible, actual conditions with real users and realistic interactions. The response times

of key elements of the system under test (e.g., web server, application server, database) are then measured by a tool and compared to pre-defined performance requirements. This can be also done for the use of memory, system input/output, CPU busy times, and access to security devices, depending on what component is (expected to be) the bottle neck or is targeted.

Based upon the analysis of results, specific elements in the architecture (hardware and software) may be modified (such as providing additional server capacity). The cycle of testing, analysis, and improvement may be repeated until the performance target is reached.

Different types of testing can be performed, depending on what needs to be measured. These include load, stress, and endurance / stability tests. Workload can be simulated by using different models: steady state, increasing, scenario-based or artificial (see [ISTQB_PT_SYL] for more details).

4.3.2 Acceptance Criteria for Performance Acceptance Tests

Performance acceptance criteria can be expressed from different perspectives as shown in the following:

- From a user perspective, the perceived response time reflects the user's real experience with the system. For example, users may abandon a web site if the response time is more than 10 seconds.
- From a business perspective, the number of concurrent users, the types of scenarios or transactions performed, and the expected response times are factors to be considered. Higher numbers of concurrent users performing resource-intensive transactions will result in longer response times. Other factors might also influence the response time based on location, time or time zone.
- From a technical perspective, available system resources (e.g., network bandwidth, CPU usage, RAM capacity) and system architecture, (e.g., server load balancing, use of data caching) are factors which influence performance efficiency. For example, web-based systems with limited network bandwidth will tend to have lower performance efficiency, especially when subjected to high loads caused by large numbers of users conducting tasks that generate significant network traffic.

The development of acceptance criteria and acceptance tests for performance requirements must address these three different perspectives (user, business and technical).

4.4 Security

Information security management and general security requirements should be part of an overall security policy for an organization (refer to the ISTQB Advanced Level Security Tester Syllabus [ISTQB_SEC_SYL] and [ISO/IEC 27005:2011] standard for further details). Business Analysts and testers should use the security policy for

recommendations and guidelines, and as a basis for managing security risks on their projects.

Security requirements should be considered at all stages of business analysis, requirements engineering and related acceptance testing including the following:

- Information security should be part of risk management and non-functional requirements elicitation and analysis. The value of information in the system under test or in a given business process should be assessed, followed by an evaluation and prioritization of security risks.
- Measurable acceptance criteria should be defined for information security requirements. They may cover a large variety of aspects such as authentication, authorization and accounting procedures, sanitization of input data, use of cryptography, and data privacy constraints.
- High-level information security test cases should be defined according to the security requirements and the acceptance criteria. These test cases define the context of the test, the main steps and the expected results.
- Some security acceptance tests can be run by the acceptance tester and others by more specialized security testers, depending on the level of technical complexity of the test.

5 Collaborative Acceptance Testing – 110 mins.

Keywords

defect, quality assurance

Learning Objectives

5.1 Collaboration

AcT-5.1.1 (K3) For a given situation, apply social and communication skills relevant for collaborative acceptance testing activities

5.2 Activities

AcT-5.2.1 (K2) Explain how to analyze discrepancies between actual and expected outcomes at the business level in a given context

AcT-5.2.2 (K2) Summarize reporting activities for acceptance testing for stakeholders

AcT-5.2.3 (K2) Explain different QA techniques for acceptance testing activities

5.3 Tool Support

AcT-5.3.1 (K1) Recall scope and objectives of tool support for acceptance testing activities

5.1 Collaboration

One challenge in acceptance testing lies in the variety and diversity of people and profiles involved (such as business analysts, technical analysts, testers, business representatives, business sponsors, automation specialists and many more), all having different objectives, different skills, and different views within the common project. Social skills are of utmost importance to gather requirements and expectations, translate them into value added solutions, align the organization and allow an effective and efficient collaboration with all stakeholders.

Both testers and business analysts need to make sure, with the help of appropriate tools and techniques (e.g., serious games, role playing, specific workshops) [Frontiera12], that team players accomplish the following:

- Get to know and understand each other to keep the team cohesion high (e.g., who's who, common goals and realizations, areas in common)
- Communicate openly in an environment of trust and respect and express their doubts, concerns or fears to identify, analyze and overcome problems (e.g., taboo-free communication, experience, perception, image)
- Envision the common objectives, the general vision, and the necessary steps to achieve them (e.g., what if it succeeds / what if it fails, what people and means are necessary, what is success)

- Defuse major problems with humor, positive communication or appropriate negotiation techniques [Ury12] (e.g., principled negotiation, caricature, role play / comics).

5.2 Activities

Defect analysis, reporting and quality assurance for acceptance testing artefacts are important and include activities that should be performed during acceptance testing.

5.2.1 Defect Analysis

Testers collect and report discrepancies between actual and expected outcome through defect reports. A defect report contains all relevant information the tester can provide to help the business analyst understand what happened and to assess the deviation.

Defect analysis is a joint activity of testers and business analysts. Usually, the tester identifies the acceptance criteria that are not satisfied. The business analyst may then be asked to analyze its impact on the related business processes. This includes determining the priority of the defect (e.g., low, medium, high, critical) with respect to its potential business impact on system usage.

To analyze the business impact of a defect, the business analyst and tester may do the following:

- Explore the path(s) in the business process models in which the defect(s) was found
- Explore the business rules which were not implemented correctly and analyze the priority of the defect from a usage point of view

The impact analysis and the resulting decision regarding further actions to be taken are documented in the defect report.

5.2.2 Reporting

Reporting activities during acceptance testing address a specific target audience (for example business managers, product managers or domain experts). These stakeholders are experts in the application domain, but they are not always familiar with implementation details. Therefore, information on acceptance test progress, results, and detected defects should be presented without technical details in the language of the target audience.

Using metrics is an important part of reporting test progress. The overall test result is provided in a test summary report. Apart from summarized information on test execution and results of all test phases, the test summary report provides additional information from the impact analysis of open defects. The test summary report also provides an indication of whether the targeted quality criteria have been reached.

Based on the test summary report, decision makers should be able to determine whether the system under test has reached the necessary pre-defined level of quality and may be released to production or not. Several outcomes are possible, including the following:

- The system may be released “as is” and rolled out without restrictions.
- The system may be released with reservations. Known errors which have a work-around are communicated to the user and the support team. For defects that represent an important risk, the related feature (or sub-system) may be excluded from roll-out.
- The system is rejected until critical defects have been fixed and re-tested. Roll-out is either postponed or replaced by alternative solutions.

5.2.3 QA Activities for Acceptance Testing

High quality acceptance tests are crucial to manage business risks associated with defects escaping to production. Quality assurance should be applied during acceptance testing activities, including the following:

- Review of acceptance criteria: The business analyst and tester verify whether the acceptance criteria are clear, consistent and comprehensive. Good acceptance criteria also cover non-functional characteristics and provide measurable pass/fail criteria.
- Review of acceptance test cases: The acceptance test cases should cover the previously defined acceptance criteria as well as business processes, business rules and business risks.
- Traceability: Traceability between requirements / user stories, acceptance criteria, test cases, and defects facilitates acceptance testing as it clarifies dependencies and provides simple access to related information.
- Coverage analysis based on traceability: If bi-directional traceability is established, it is possible to perform a systematic coverage analysis.
- Review of test reports: Test reports should be clear, consistent and comprehensive. They should contain all information provided by the tester to support decisions about the release.

5.3 Tool Support

The tooling for acceptance testing activities originates from both the business analysis and software testing domains.

The following table lists some of the types of tools that can be used in acceptance testing activities.

Tool Type	Usage for Acceptance Testing
Requirements management tool	<ul style="list-style-type: none">• Description of acceptance criteria• Traceability between tests and requirements• Coverage analysis

Agile project management tool	<ul style="list-style-type: none">• Description of acceptance criteria• Traceability between tests and user stories• Coverage analysis
Business process management tool	<ul style="list-style-type: none">• Model business process and rules• Analyze defect impact on business processes
Test management and automation tool	<ul style="list-style-type: none">• Manage acceptance tests and test execution campaign• Manage test execution results
Model-based testing tool	<ul style="list-style-type: none">• Generate test cases from business process models• Manage traceability between business process models, business rules, requirements and test cases
Defect / Incident management tool	<ul style="list-style-type: none">• Manage defect / incident lifecycle

6 Abbreviations

Abbreviation	Meaning
ATDD	Acceptance Test-Driven Development
BA	Business Analyst
BDD	Behavior Driven-Development
BPMN	Business Process Model and Notation
CTFL	Certified Tester Foundation Level
COTS	Commercial Off-The-Shelf software
CPU	Central Processing Unit
DMN	Decision Model and Notation
IQBBA®	International Qualification Board for Business Analysis
ISO	International Organization for Standardization
ISTQB®	International Software Testing Qualifications Board
OMG	Object Management Group
OAT	Operational Acceptance Testing
QA	Quality Assurance
UAT	User Acceptance Testing
UX	User eXperience

7 Registered Trademarks

Trademark	Owner
BPMN™	Object Management Group, Inc.
CTFL®	ISTQB®
DMN™	Object Management Group, Inc.
IQBBA®	International Qualification Board for Business Analysis
ISTQB®	International Software Testing Qualifications Board

8 References

Standards

[ISO/IEC 25010:2011] Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Software and quality in use models, 2011.

[ISO/IEC 29148:2011] Systems and software engineering - Life cycle processes - Requirements engineering, 2011.

[ISO/IEC 27005:2011] ISO/IEC 27005:2011 Information technology - Security techniques - Information security risk management, 2011

[ISO/IEC 9241-11:1998] ISO/IEC 9241-11:1998 - Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability, 1998

[ISO/IEC 19510:2013] ISO/IEC 19510:2013 - Information technology - Object Management Group Business Process Model and Notation

[OMG BPMN 2.0] OMG BPMN 2.0 standard documentation - January 2011
<http://www.omg.org/spec/BPMN/2.0/>

[OMG DMN 1.2] OMG DMN 1.2 standard documentation - January 2019
<https://www.omg.org/spec/DMN/1.2/>

IQBBA® Documents

[IQBBA_GLOSSARY] Standard glossary of terms used in Software Engineering, version 1.0 – IQBBA® 2011

[IQBBA_FL_SYL] Certified Foundation Level Business Analyst (CFLBA) Syllabus, version 3.0 – IQBBA® 2018

ISTQB® Documents

[ISTQB_FL_SYL] ISTQB® Certified Tester Foundation Level Syllabus, Version 2018

[ISTQB_FL_AT_SYL] ISTQB® Agile Tester Foundation Level Syllabus, Version 2014

[ISTQB_MBT_SYL] ISTQB® Foundation Level Model-Based Tester, Version 2015

[ISTQB_UT_SYL] ISTQB® Foundation Level Usability Testing, Version 2018

[ISTQB_PT_SYL] ISTQB® Foundation Level Performance Testing, Version 2018

[ISTQB_SEC_SYL] ISTQB® Advanced Level Security Tester, Version 2016

[ISTQB_GLOSSARY] Standard Glossary of Terms used in Software Testing, Version 3.2, 2018

Referenced in this Syllabus

[Anderson01] Lorin W. Anderson, David R. Krathwohl (eds.) "A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives", Allyn & Bacon, 2001, ISBN 978-0801319037

[Cohn04] Mike Cohn, "User Stories Applied: For Agile Software Development", Addison-Wesley Professional, 2004, ISBN: 978-0321205681

[Frontiera 2012] Joe Frontiera and Daniel Leidl, "Team Turnarounds: A Playbook for Transforming Underperforming Teams", Jossey-Bass; 1st edition (July 24, 2012), ISBN: 978-1118144787

[Nielsen94] Jakob Nielsen "Heuristic evaluation". Nielsen, J., and Mack, R.L. (Eds.), "Usability Inspection Methods" - John Wiley & Sons, New York, NY, 1994, ISBN 0-471-01877-5

[Pugh11] Ken Pugh, "Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration", Addison-Wesley Professional, 2011, ISBN: 978-0321714084

[Rubin08] Jeffrey Rubin and Dana Chisnell, "Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests", Wiley; 2nd edition (May 12, 2008), ISBN: 978-0470185483

[Smart14] John Ferguson Smart, "BDD in Action: Behavior-driven development for the whole software lifecycle", Manning Publications, 2014, ISBN: 978-1617291654

[Ury12] Roger Fisher and William Ury, "Getting to yes", Patton Ed. Random House Business Books, 2012, ISBN: 978-1847940933

[Whittaker09] James Whittaker, "Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design", Addison-Wesley Professional; 1st edition (September 4, 2009), ISBN: 978-0321636416

Appendix A – Subset of BPMN 2.0.1 and DMN 1.2

This syllabus references and uses the following versions of both OMG standards:




- BPMN 2.0.1 published in 2011, and ratified as ISO 19510 in 2013
- DMN 1.2 published in 2019

A.1 Subset of BPMN 2.0.1

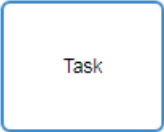
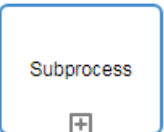
BPMN models consist of simple diagrams constructed from a limited set of graphical elements. Four basic element categories are provided: flow object, connecting objects, swim lanes and artifacts. For each of these categories, the following sections present the exact subset of graphical elements that pertain to this syllabus; these are to be used for the purpose of K3 level learning objectives (in Chapter 3).

Flow Objects

Events:



Start	Intermediate	End
		

Activities:

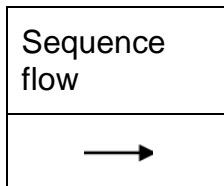
Task	Subprocess
	

Four types of tasks are considered in the syllabus: undefined, service, user and business rule tasks.

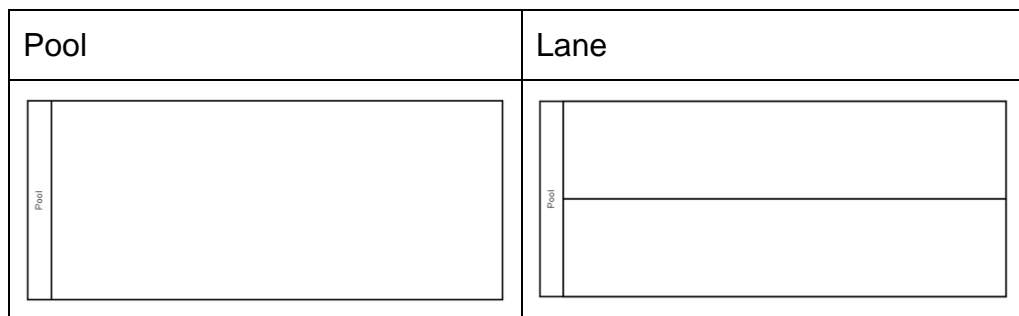
Gateways:

Exclusive	Parallel
	

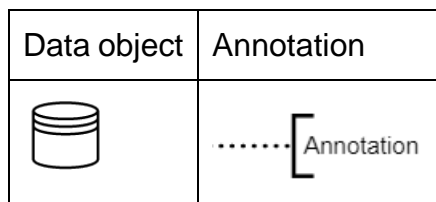
Connecting objects



Swim lanes:



Artifacts:



For the use of this subset of BPMN graphical elements, syntactic, semantic and pragmatic rules are those defined within the BPMN 2.0.1 standard [OMG BPMN 2.0] [ISO/IEC 19510:2013].

For this syllabus, only Private (internal) business processes are applicable to describe the workflows to be tested during the acceptance testing activities.

A.2 Subset of DMN 1.2

DMN models consist of simple diagrams constructed from a limited set of graphical elements, and decision tables. The diagrams support the representation of decision requirements, and the decision tables represent the related decision logic. A declarative language is also defined within the standard to allow a formal definition of decisions. In this syllabus, only those decision tables using DMN 1.2 notation are applicable to represent business rules linked with workflows represented in BPMN 2.0 (see the previous section).

A decision table consists of [OMG DMN 1.2]:

- An information item name
- A list of input clauses (zero or more)
- A list of output clauses (one or more)
- A set of outputs (one or more)
- A list of annotation clauses (zero or more)
- A list of rules (one or more)

DMN decision tables can be connected to BPMN business process models by using business rule tasks (see Appendix A1).