IQBBA® & ISTQB®
Certified Foundation Level
Acceptance Testing

Syllabus

Version 2018
Beta Version

International Qualification Board for Business Analysis

International Software Testing Qualifications Board

Acceptance Testing Working Group: Olivier Denoo, Debbie Friedenberg, Anne Kramer, Bruno Legeard (chair), Karolina Zmitrowicz; 2018.

Revision History

Version	Date	Remarks
Beta	August 24th, 2018	Beta version for GA Berlin – Comments from the cross review have been managed.
Alpha 1.1	May 28th, 2018	updated after finalization of sample exam
Alpha 1.0	November 10th, 2017	

Table of Contents

Acknowledgements

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

The team thanks the following persons from the IQBBA® and ISTQB® National Boards, who participated in reviewing, commenting, and balloting of this Syllabus: Chris Van Bael, Graham Bath, Ernst von Düring, Beata Karpinska, Ingvar Nordström, Monika S. Olsen, Lara Pellegrino, Jan Sabak, Lucien Stapp, TBC.

This document was formally approved for release by the IQBBA® on XXXX and by the ISTQB® on YYYY.

0 Introduction to this Syllabus

0.1 Purpose of the Document

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

The IQBBA and ISTQB provide this syllabus as follows:

1. 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.
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 IQBBA and ISTQB may allow other entities to use this syllabus for other purposes, provided they seek and obtain prior written permission.

0.2 Overview

The IQBBA and ISTQB Certified Foundation Level Acceptance Testing syllabus is a specialist module at the Foundation Level.

0.3 Learning Objectives

The Learning Objectives support the Business Outcomes and are used to create examinations for achieving the IQBBA and ISTQB Foundation Level Acceptance Testing certification.

In general, all parts of this syllabus are examinable at a K1 level, i.e., the candidate will recognize, remember and recall terms and concepts stated in the syllabus. The specific Learning Objectives at K1, K2 and K3 levels are indicated at the beginning of each chapter within this syllabus.

1 Introduction and Foundations – 75 mins.

Keywords: acceptance criteria, acceptance testing, requirement, user story

Learning Objectives:

1.1 Fundamental Relations

AcT-1.1-1 (K1) Recall the relation between business goals, business needs and requirements

AcT-1.1-2 (K2) Explain the relation between requirements / user stories, acceptance criteria and acceptance test cases

AcT-1.1-3 (K2) Explain how the quality of requirements affects acceptance testing

1.2 Business Analysis and Acceptance Testing

AcT-1.2-1 (K2) Summarize the relation between acceptance testing activities and business analysis activities

AcT-1.2-2 (K1) Recall how acceptance testing relates to other test levels

AcT-1.2-3 (K2) Explain how Testers and BAs collaborate in acceptance testing activities

AcT-1.2-4 (K1) Recognize how acceptance testing is used in ATDD and BDD

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, and work done by one group may greatly effect, either positively or negatively, that of the other. This is especially true in the area of 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 release draws near. By taking advantage of the complementary skills and by working together, business analysts and testers can positively affect the development process, 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.

1.1 Fundamental Relations

1.1.1 Business Goals, Business Needs and Requirements

Business analysts first have to understand the organization's overall business goals and identify current business processes. Then, they describe specific business needs and determine a business case, which 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 describe at different levels of abstraction, what shall be achieved. In Agile development processes such as Scrum, the same principles apply, but different terms may be used (for example epics, features and user stories).

Hint: 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 Test Cases

During requirements elicitation, testers and business analysts should begin to create specific acceptance criteria and develop acceptance testing as a joint effort, so that right from the beginning of the project, there is a mutual understanding of what “acceptance” means from the business, the development and the testing perspective.

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 refine the story.

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 confirm whether a requirement or user story has been implemented as expected. Thus, acceptance criteria provide the test conditions for acceptance test cases, but do not contain the detailed test procedure.

Acceptance test cases are based on acceptance criteria. They specify how the verification of the acceptance criteria shall be performed.

1.1.3 The Importance of the Quality of Requirements

If acceptance criteria and tests are based on requirements that are vague or ambiguous, it is likely that testers will make assumptions about stakeholder expectations and business needs. In this case, it is possible that they write incorrect acceptance criteria, and that the resulting tests will be flawed. This will lead to rework or, even worse, the running of invalid tests, thus creating unnecessary costs.

For this reason, it is critical for testers to review requirements and 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 criteria and tests are valid and are a meaningful way to determine a product’s readiness for release.

In Agile development, the INVEST technique [INVEST] defines quality characteristics for user stories, that may be used by BAs / product owners, developers and testers to ensure the quality of user stories (cf. ISTQB Foundation Level Agile Tester syllabus [ISTQB-FL-AGILE]).

1.2 Business Analysis and Acceptance Testing

1.2.1 Relationship between Business Analysis and Testing Activities

The main elements of the IQBBA Business Analysis activities are [IQBBA_FL]:

- Strategy definition
- Management of Business Analysis process
- Requirements Engineering in Business Analysis
- Solution evaluation and optimization

A test process consists of the following main groups of activities (cf. ISTQB Certified Tester Foundation Level syllabus [ISTQB-FL-SYL]):

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

Quite a number 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:

IQBBA Business analysis / ISTQB Test planning, Test analysis and Test design:
During the IQBBA business analysis activities, business analysts prepare detailed business requirements and acceptance criteria. These correspond to the work being done during the ISTQB 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 (including acceptance test cases). Testers can contribute to the definition and verification of acceptance criteria as part of acceptance testing activities. Working together, the two roles ascertain that there is proper understanding of the business solution and agree on the appropriate level of testing.

IQBBA Solution evaluation / ISTQB Test implementation, Test execution and Test completion:
During the IQBBA solution evaluation phase, business analysts work with testers during the corresponding ISTQB test implementation and test execution activities to review the testers' test cases, clarify issues and potentially help with creation of test data to support on of business related tests. They can assist with the preparation and by participating in acceptance testing. Business analysts may also support testers by evaluating test results. In addition, business analysts may assist testers in the test completion activities.

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

1.2.2 Acceptance Testing within the Test Levels

In Agile development, acceptance test cases are developed for user stories selected for the current iteration and run during the iteration. Iteration-level acceptance testing may then be automated and become part of the overall regression test suite for the project.

In sequential development, acceptance test execution generally takes place at the end of the project by using the fully functional, fully tested product in order to do a final business assessment on its readiness for release.

If the overall system consists of multiple integrated products or sub-systems, acceptance testing may be appropriate at the system integration test level. This is especially true in cases where one or more of these sub-systems was developed externally and has been customized.

As stated in [ISTQB_FL_SYL], acceptance testing typically focuses on the behavior and capabilities of a whole system or product. Common forms of acceptance testing include user acceptance testing, operational acceptance testing, contractual and regulatory acceptance testing, alpha and beta testing. This acceptance testing syllabus is focused on general acceptance testing practices that can be used in these different forms rather than a detailed and specific description of each form.

1.2.3 Collaboration between BAs and Testers in Acceptance Testing

Given the unique position these two roles of BAs and testers have in the organization, there are a variety of opportunities for the two to collaborate during acceptance testing activities:

- review and discuss specifications and requirements to ensure that appropriate acceptance criteria are provided
- collaborate on test planning and risk analysis to ensure that further on, appropriate test cases are developed and prioritized
- jointly discuss, create and review test conditions and test cases
- work together on execution of acceptance test cases
- collaborate on analyzing test results

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 formalized testing, the more likely the two groups will understand and appreciate each other allowing them to better collaborate and work effectively within the project team.

1.2.4 How Acceptance Testing Can Drive the Development Process

The wide acceptance of Agile software development practices has influenced how acceptance testing relates to requirements elicitation and other business analysis activities. Instead of considering acceptance testing as an activity to be handled by the testers after requirements have been finalized, acceptance criteria and acceptance test cases should be created much earlier to have a greater impact on the overall development of the solution.

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

- In Acceptance Test Driven Development (ATDD) [Pug11], acceptance test cases are produced collaboratively during requirements analysis by business analysts, product owners, testers and developers.
- Behavior Driven Development (BDD) [Sma14] uses domain-specific scripting language, based on natural language statements, for the creation of acceptance test cases at early stage and to facilitate their automation for test execution.

Both approaches:

- involve the entire agile team and help to focus the development efforts on the business goals.
- see acceptance test cases as living documentation of the product.

2 Acceptance Criteria, Acceptance Test Cases and Experience Based Practices – 165 mins.

Keywords: acceptance criteria, beta testing, exploratory testing, test case

Learning Objectives:

2.1 Writing Acceptance Criteria

AcT-2.1-1 (K3) For a given requirement or user story, write a set of acceptance criteria that meet good practices

2.2 Designing Acceptance Test Cases

AcT-2.2-1 (K2) Explain test approaches and test techniques for acceptance testing

AcT-2.2-2 (K3) Design acceptance test cases for a given user story using the Gherkin language

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 relation between beta testing and acceptance testing

Specifying acceptance criteria is an important acceptance testing activity. 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 of the acceptance testing phase, and increases the chance of a successful iteration or product release.

2.1 Writing Acceptance Criteria

Writing acceptance criteria forces business analysts and testers to think about the functionality from a stakeholder's or from a usage perspective. This supports the 'shift left testing' principle and allows an early validation of the related requirement or user story, because inconsistencies, contradictions and missing information will have a better chance to be detected. Acceptance criteria do not contain solution details. It is possible to specify them without knowing implementation details.

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 decide whether or not the test object satisfy 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 is it achieved?".
- Acceptance criteria should address non-functional aspects (quality characteristics) as well as functional requirements.

As requirements and user stories, acceptance criteria should be reviewed for example through walkthrough or technical review.

2.2 Designing Acceptance Test Cases

2.2.1 Test techniques for acceptance testing

By definition, acceptance testing is a requirements-based approach to testing. The tester derives acceptance test cases from the acceptance criteria related to each requirement / user story using specification-based black-box test techniques (for example equivalence partitioning, boundary value analysis or use case testing – (see [ISTQB_FL_SYL])).

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 "Derive Acceptance Test Cases from Business Process Models").
- Experience-based testing takes advantage of the tester's experience, knowledge and intuition (see section 2.3.1 "Exploratory Testing")
- Model-based testing approaches, using models to derive test cases [ISTQB_FL_MBT_SYL]

In a risk-based testing approach, prioritization and intensity of testing depends on previously identified product risks. In a model-based testing approach, the acceptance test procedure is derived from graphical (or textual) models. In all approaches, the expected results defined in the acceptance test cases are derived from acceptance criteria.

2.2.2 Using the Gherkin Language to write Acceptance Test Cases

In Behavior Driven Development, acceptance test cases are often formulated in a semi-formal language, referred to as Gherkin language [Sma14]. Using the Gherkin language, acceptance 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 even other stakeholders to write test scenarios in a way that is easily shared by stakeholders. Acceptance test cases written in Gherkin language do not refer to user interface elements in the "When" block. Instead, they describe what the tester wants to achieve with the action. The expected result in the "Then" block corresponds to the applicable acceptance criterion.

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

2.3.1 Exploratory Testing

In the context of acceptance testing, exploratory testing [Whi09] is an experience-based test technique that does not make use of detailed predefined test procedures. The testers are domain experts, but are not necessarily familiar with the product under test.

During an exploratory testing session, the tester:

- learns how to work with the product,

- designs the tests,
- performs the tests and
- interprets the results

all within the same session, that is, an uninterrupted period of time spent in executing tests.

It is good practice in exploratory testing to use an exploratory test charter. The test charter is prepared prior to the testing session, possibly jointly between the BA and the tester and is used by the tester during the session. It includes information on the role the tester takes during the session, the particular objective he wants to achieve during the session, the setup, the activities that would be interesting to test, the test oracle and other relevant information [ISTQB_AT_FL_SYL]. Time-boxed sessions help to control the time and effort dedicated to exploratory session.

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 this iteration.

Exploratory testing should be used to complement other approaches in acceptance testing, for example to obtain provide rapid feedback on new features.

2.3.2 Beta Testing

Beta testing is a form of acceptance testing, often used for commercial off-the-shelf software (short: COTS) or software as a service platform. It is conducted to acquire feedback from the market.

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

During beta testing, may be discovered defects in the product that escaped during the development process and previous test levels. The product is tested in various realistic configurations by genuine users in their business process context. They help organizations to avoid costly hotfixes or product recalls on a larger scale. Therefore, in many context, acceptance testing should include beta tests as well as acceptance test scenarios based on acceptance criteria.

Acceptance testing should not be limited to beta testing, because beta testing is not systematic and it is neither measurable nor guaranteed that all requirements or user stories are covered by the tests. Moreover, beta testing is performed late in the development process whereas writing acceptance criteria supports the early testing principle.

3 Business Process and Business Rules Modeling – a Common Language between BAs and Testers – 135 mins.

Keywords: business process-based testing

Learning Objectives:

3.1 Modeling Business Processes and Rules

AT-3.1-1 (K3) Draw a simple business process/rule model using BPMN and or DMN notations

3.2 Deriving Acceptance Test Cases from Business Process/Rule Models

AT-3.2-1 (K3) From a given and simple business process/rule model (in BPMN and or DMN), derive a set of test scenarios covering a given coverage criterion

3.3 Good Practices of Business Process Modeling for Acceptance Testing

AT-3.3-1 (K2) Summarize the good practices of business process and business rules modeling for acceptance testing

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

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 should be an essential part of acceptance testing goals. BPMN and DMN provide a common language for business analysts and testers for graphically representing business processes and business rules in a standard way. This representation helps to prioritize the most important tests to execute. It also supports the design and implementation of those tests.

3.1 Modeling Business Processes and Rules

Representing business processes and rules to be tested using 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 make explicit criterion for guiding behavior, shaping judgments, or taking decisions.

Business Process Model and Notation (BPMN), maintained by the Object Management Group (OMG), is a recognized standard for business process modeling which uses a flowcharting technique. In this syllabus, a subset of the BPMN 2.0 notation is used, sufficient to draw simple business process models in the context of acceptance testing activities. This subset is defined in Appendix A1.

Decision Model and Notation (DMN), also standardized by the OMG, is complementary to the BPMN standard. While BPMN is used to represent workflows, DMN is used to represent decisions and business rules within the workflow. In this syllabus, a subset of the DMN 1.1 notation is used, 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 Test Cases from Business Process/Rule Models

A business process model with business rules, described with the BPMN 2.0 and or DMN 1.1 notations, provides a precise definition of the scenarios to be tested including the cases related to business rules. Therefore, the business process models provide a basis to generate relevant acceptance test cases using a business process-based testing approach.

Typical coverage criteria for the business process models when generating acceptance test cases are:

- coverage of user stories, requirements & risks annotated in the business process model
- coverage of decisions in the decision tables describing the business rules
- coverage of all paths without loops in the business process diagram
- coverage of user scenarios based on the walkthrough in the business process model

These coverage criteria may be combined, with the goal to meet the acceptance testing objectives.

3.3 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:

- Graphical representations of business processes in BPMN should focus on what needs to be tested. Therefore, workflow descriptions that cover only partially the behavior of related software systems are acceptable, as long as they represent what is to be tested.
- Using decision tables in DMN allows defining test conditions corresponding to the business rules under test.
- Diagrams should be as simple as possible and be structured in sub-processes when needed in order to limit the number of graphical elements in a single business process diagram.
- Business process modeling for acceptance testing should be a collaborative work between business analysts and testers, and produced artifacts should be shared between both roles.
- 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.

4 Acceptance Testing for Non-Functional Requirements – 120 mins.

Keywords: usability, user experience, performance efficiency, security

Learning Objectives:

4.1 Quality attributes

AcT-4.1-1 (K1) Recall non-functional characteristics for system/software product quality according to ISO/IEC 25010

4.2 Usability and User eXperience

AcT-4.2-1 (K2) Compare the concepts of usability and user experience and illustrate their differences with practical examples

AcT-4.2-2 (K2) Exemplify how different users interact with software in various contexts and how this can impact acceptance testing

AcT-4.2-3 (K2) Map different types of usage scenarios to the four pillars of UX requirements analysis (user analysis, task analysis, context analysis and competition analysis)

AcT-4.2-4 (K2) Summarize different methods for testing usability and identify their respective domains of application

4.3 Performance

AcT-4.3-1 (K2) Exemplify high-level performance scenarios according to given efficiency requirements

AcT-4.3-2 (K2) Explain the impact of different perspectives on performance efficiency

4.4 Information security

AcT-4.4-1 (K2) Exemplify information security acceptance criteria and related high-level security test scenarios that are required for a project in accordance with a given security requirement

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

4.1 Non-functional characteristics

Non-functional characteristics strongly influences user acceptance of a proposed solution. Even if their criticality depends on the context, not properly addressing them usually results in severe issues (dissatisfaction, lost sales, rejection of the solution, liability risks, public exposure of the organization and other problems).

The ISO 25010 standard [ISO 25010:2011] introduces a system and software product quality model that categorizes product quality properties into the following eight characteristics:

- functional suitability
- reliability
- performance efficiency

- usability
- security
- compatibility
- maintainability
- portability

Each characteristic is composed of a set of related sub characteristics presented in Appendix B.

Except for functional suitability, all the characteristics introduced in the product quality model define non-functional quality attributes. An acceptance testing strategy should provide priorities for non-functional characteristics to be tested by acceptance tests.

4.2 Usability and User eXperience

4.2.1 Concepts of usability and user experience

In software engineering, usability is the degree to which a software product can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a specified context of use [ISO/IEC 9241-11:1998]. It can be measured against objectives (e.g. learnability, efficiency, memorizing, satisfaction, error prevention) mostly in the form of heuristics [Rub08][Nie94].

The lack of usability results in frustration, refusal to use the software and, in the most critical instances, in injuries or death of the user.

User eXperience (abbrev. UX) has a broader scope for it considers notions like usefulness, utility as well as the user's emotions or attitude towards it. Context has a strong influence as our experience of a software or service may totally differ whether we are sitting at a desk, driving a car or hiking; depending on weather (e.g. sun, rain, cold) or health conditions (e.g. fatigue, age); stressful or noisy environments and other aspects.

Further details regarding usability testing are provided in the ISTQB Foundation Level Usability Testing syllabus [ISTQB-FL-UT]

4.2.2 Usability and Accessibility Limitations

Due to common human limitations and biases (e.g. cognitive or perceptive biases, visual impairment or inexperience) 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 ensure that products or services are accessible to all users being stakeholders for the solution by considering these limitations when designing acceptance testing criteria and test cases.

4.2.3 User eXperience requirements analysis

User experience requirements analysis is based upon four major pillars:

- User analysis: Users are qualified in terms of physical and intellectual characteristics, technical skills, business knowledge, socio-economic and cultural background. Business analysts can also use models (e.g. based on personas).

- Tasks 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 (such as light, temperature, movement, humidity or dust), physical conditions (e.g. sitting, standing, lying, moving, hands-free) or "psychological" conditions (such as 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 competition and take inspiration from the successful implementation of their solutions in order to retain or attract users and customers. Another source of inspiration can come from successful solutions found in similar or even different sectors.

4.2.4 Usability testing

There are different approaches to testing usability:

- Walkthrough and thinking aloud methods: users explore the product or systems and describe their actions and impressions out loud while doing so. They may be given specific tasks to accomplish in order to identify how they interact with the product and to learn about expectations or difficulties.
- Checklist-based evaluations: users evaluate the system or product under test according to standardized checklists [Rub08] to evaluate, compare and qualify their experience.
- Biometrics-based evaluations: user behavior is monitored with specific biometric devices (eye-movement recording, mouse-eye-movement recording to understand how the user interacts with a page or a system, what attracts their attention, what is more or less visible.)
- 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.
- Other approaches: like log files analysis. The goal is to analyze retrospectively how the users interacted with the system to improve it.

4.3 Performance Efficiency

4.3.1 High-level performance acceptance test cases

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 close as possible, actual conditions with real users and realistic interactions. The response time 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.

Based upon the analysis of results, specific elements in the architecture (hardware and software) may be modified (e.g., 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.

4.3.2 Test conditions for performance acceptance test cases

Performance test conditions can be expressed in terms of different perspectives:

- From a user perspective, the perceived response time reflects his real experience with the system. As a rule of thumb, for example, users typically abandon a web-site if the response time is more than a couple of seconds.
- From a business perspective, the number of concurrent users, the types of scenarios or transactions performed, and the expected response time are factors to be considered. Generally speaking, higher numbers of concurrent users performing resource-intensive transactions will result in longer response times Some extra elements might also influence the response time (e.g. location, time or time zone).
- From a technical perspective, available system resources (e.g., network bandwidth, CPU availability, 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 resource-intensive tasks.

Therefore, good collaboration between the parties involved together with a strong understanding of their respective needs is a key to success.

4.4 Information 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-AL-SEC] and [ISO/IEC 27005:2011] standard for further details. Therefore, business analysts and testers should use this policy for recommendations and guidelines, and as a basis for managing security risks on their projects.

In this context, information security should be considered at all stages of business analysis, requirements engineering and related acceptance testing:

- Information security should be part of risk management and non-functional requirements elicitation and analysis. This means that an assessment of the value of information for the system under test or a given business process should first be done, 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 at authentication, authorization and accounting procedure, sanitization of input data, use of cryptography, or data privacy constraints.
- High level information security test scenarios should be defined according to the security requirements and the acceptance criteria. These high level security test scenarios define the context of the test, the main steps and expected results.
- Some security acceptance tests can be run by the acceptance tester and others by more specialized security testers depending the level of technical complexity for the test.

5 Collaborative Acceptance Testing – 110 mins.

Keywords: Incident report

Learning Objectives:

5.1 Collaboration

AcT-5.1-1 (K3) For a given situation, practice 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 outcome at business level in a given context

AcT-5.2-2 (K2) Summarize reporting activities for acceptance testing for stakeholders

AcT-5.2-3 (K2) Exemplify 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

One of the big difficulties in acceptance testing lies in the variety and diversity of people and profiles involved (i.e. 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.

5.1 Collaboration

Testers (as well as Business Analysts) need to make sure, with the help of appropriate tools and techniques (e.g. serious games, role plays, specific workshops) [Fron12], that team players:

- get to know and understand each other to keep the team cohesion high (e.g. who's who, common goals and realizations, what have we got 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, catharsis, perception, image...)
- envision the common objectives, the future of the project and the necessary steps to reach it (e.g. what if it succeeds / what if it fails, what people & means are necessary to reach it)
- defuse major problems with humor, positive communication or appropriate negotiation techniques [Ury12] (e.g. principled negotiation, caricature, role play / comics).

5.2 Activities

Incident analysis, change management, reporting and quality assurance for acceptance testing artefacts are important activities that should be achieved during acceptance testing.

5.2.1 Incident analysis

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

If the incident found during acceptance testing leads to be a defect, the business analyst may be asked to analyze its impact on the related business processes. This includes determining the severity of the defect (e.g. low, medium, high, critical) with respect to its potential business impact on system usage. Testers may be involved in incident analysis too.

To analyze the business impact of a defect, the business analyst may:

- identify the acceptance criteria that are not satisfied
- explore the path(s) in business process models than are interrupted and the specific business rules that are not correctly implemented
- analyze the likelihood and impact 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 incident report.

5.2.2 Reporting

Reporting activities during acceptance testing address a specific target audience: business domain manager, product and project manager, domain expert and others. These stakeholders are experts in the application domain, but they are usually not familiar with implementation details. Therefore, information on acceptance test progress, results and detected defects should be presented in a condensed manner in the language of the target audience.

Using metrics is an important part of reporting test progress. The overall test result is given in a test summary report. Apart from synthetized information on test execution and results, the test summary report provides additional information coming from impact analysis and risk assessments of reported defects. The test summary report also provides an indication if the quality criteria described in the contract are reached.

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

- The system is accepted “as is” and rolled out without restrictions.
- The system is accepted with reservations, but some features (or sub-systems) are excluded from roll-out, because the defects found in those parts represent an important risk.
- 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 test cases are crucial to control business risks associated with defects escaping to production. Therefore, several quality assurance techniques are applied during acceptance testing activities:

- Review of test acceptance criteria

Business analyst and tester verify whether the acceptance criteria are clear, consistent and comprehensive. Good acceptance criteria include non-functional quality criteria and provide measurable pass/fail criteria.

- Review of acceptance test cases
  The acceptance test cases shall 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 considerably facilitates acceptance testing tasks, as it clarifies dependencies and provides simple access to related information.
- Coverage analysis based on traceability
  If traceability is established, it is possible to perform a systematic coverage analysis.

Tool support exists to perform automated coverage analysis based on traceability information.

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 various types of tools useful for acceptance testing activities:

Type of tools	**Usage for acceptance testing**
Requirements management software	• description of acceptance criteria • traceability between acceptance test cases and requirements
Agile project management software	• description of acceptance criteria • traceability between acceptance test cases and user stories
Business process management software	• model business process and rules • analyze defect impact on business processes
Test management and automation software	• manage acceptance test cases and test execution campaign • manage test execution results
Business process-based test design and implementation tool	• generate tests from business process models • manage traceability between business process models, business rules, requirements and test cases
A/B testing tool	• support Alpha/Beta testing
Defect / Incident management tool	• manage defect / incident lifecycle

6 Abbreviations

Abbreviation	Meaning
A/B	Alpha / Beta
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
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
QA	Quality Assurance
UX	User eXperience

7 Registered Trademarks

Trademark	Owner
BPMN™	Object Management Group, Inc.
DMN™	Object Management Group, Inc.

8 References

Standards

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

[OMG DMN 1.1] OMG DMN 1.1 standard documentation - http://www.omg.org/spec/DMN/1.1/

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

IQBBA Documents

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

[IQBBA_FL] 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_AT_FL_SYL] ISTQB Agile Tester - Foundation Level Syllabus, Version 2014

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

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

Referenced in this Syllabus

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

[INVEST] Bill Wake, “INVEST in Good Stories, and SMART Tasks,” http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ .

[Fron12] Joe Frontiera and Daniel Leidl, “Team Turnarounds: A Playbook for Transforming Underperforming Teams Hardcover”, Jossey-Bass; 1 edition (July 24, 2012), ISBN: 978-1118144787

[Nie94] “Heuristic evaluation”. In Nielsen, J., and Mack, R.L. (Eds.), “Usability Inspection Methods” - John Wiley & Sons, New York, NY; 1994

[Pug11] Ken Pugh, “Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration”, Addison-Wesley Professional, 2011, ISBN: 978-0321714084

[Rub08] Jeffrey Rubin and Dana Chisnell, “Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests”, Wiley; 2 edition (May 12, 2008), ISBN: 978-0470185483

[Sma14] 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

[Whit09] James Whittaker, “Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design”, Addison-Wesley Professional; 1 edition (September 4, 2009), ISBN: 978-0321636416

Appendix A – Subset of BPMN 2.0 and DMN 1.1

This syllabus references and uses the following version of both OMG standards: BPMN 2.0 published in 2011 and DMN 1.1 published in 2016.

A.1 Subset of BPMN 2.0

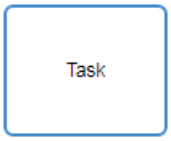
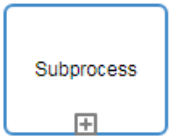
BPMN models consist of simple diagrams constructed from a limited set of graphical elements. Four basic element categories are provided: flow object, connecting objects; swim lanes and artifacts. For each of these categories, the following sections present the exact subset of graphical elements that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

Flow objects

Events:

Start	Intermediate	End
		

Activities:

Task	Subprocess
	

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

Gateways:

Exclusive	Parallel
	

Connecting objects

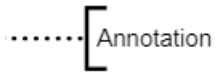
Sequence flow


Sequence flow, message flow, association

Swim lanes:

Pool	Lane
	

Artifacts:

Data object	Annotation
	

For the use of this subset of BPMN graphical elements, syntactic, semantic and pragmatic rules are those defined within the BPMN 2.0 standard [BPMN 2.0].

For this syllabus, only Private (internal) business processes [BPMN 2.0] are applicable to describe the workflows to be tested during the acceptance testing activities.

A.2 Subset of DMN 1.1

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 1.1 notation are applicable in order to represent business rules linked with workflows represented in BPMN 2.0 (see the previous section).

A decision table consists of [OMG DMN 1.1]:

- An information item name.
- An output label.
- A set of inputs (zero or more).
- A set of outputs (one or more).
- A list of rules (one or more).

Connecting BPMN 2.0 business process models and DMN 1.1 decision tables can be connect to BPMN 2.0 business process models by using business rule tasks (see Appendix A1).

Appendix B – ISO/IEC 25010:2011 Product Quality Characteristics and Subcharacteristics

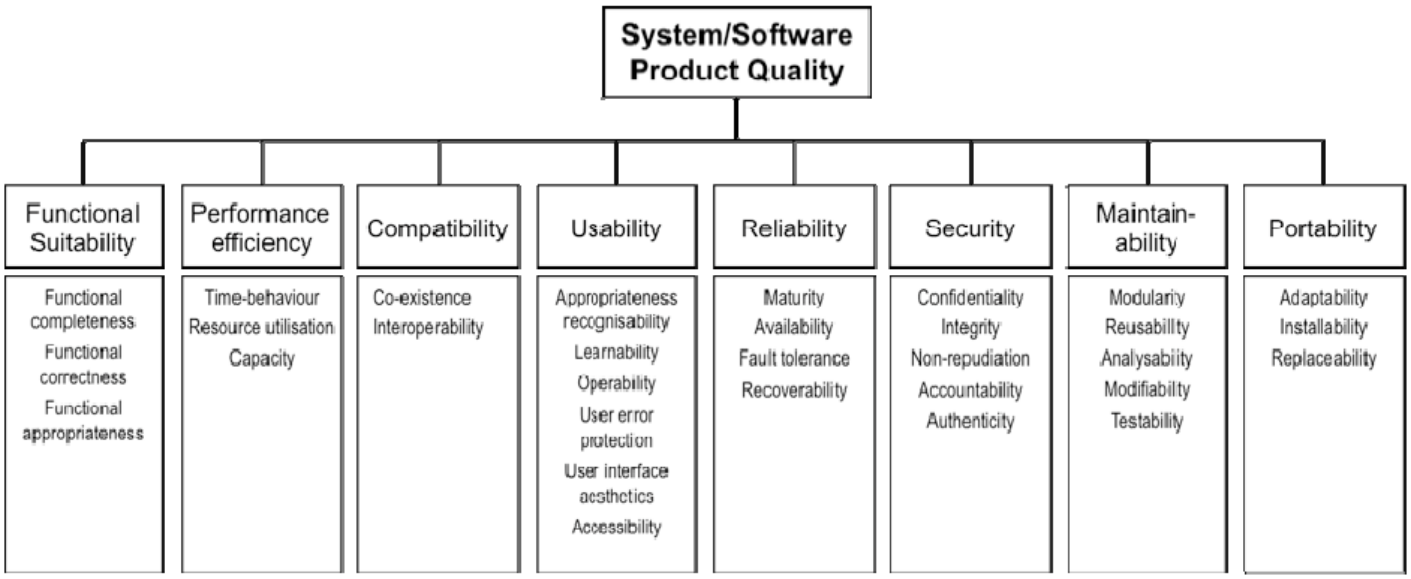


Figure 1 - ISO/IEC 25010:2011 Product quality model