DELIVERABLE

Project Acronym: **ANTILOPE**

Grant Agreement number: **325077**

Project Title: **ANTILOPE - Adoption and take up of standards and profiles for eHealth Interoperability"**

D2.2: Interoperability Testing Processes

Revision: 1.0

Authors:

Morten Bruun-Rasmussen, (in cooperation with MedCom)
Ib Johansen, MedCom

Review:

Catherine Chronaki, HL7 Foundation
Gerard Freriks, EN 13606 Association
Stephen Kay, CEN

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination level		
P	Public	✓
C	Confidential, only for members of the consortium and the Commission Services	

REVISION HISTORY AND STATEMENT OF ORIGINALITY

Revision History

Revision	Date	Author	Organisation	Description
0.1	10.07.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Draft based on HITCH D1.2 Profile QMS description
0.2	23.07.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Update with comments from Vincent van Pelt NICTEZ
0.3	23.07.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Update with comments from Barry Reinhold, Continua and Gerard Freriks CEN 13606
0.8	18.07.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Update + adding chapter 1 + 2.
0.81	27.10.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Executive summary added
0.9	21.11.13	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Update – using the new Antilope deliverable template
0.91	20.09.14	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Update with comments from EC annual project review
0.99	28.11.14	Morten Bruun-Rasmussen	(in cooperation with MedCom)	Review and comments from SEP. Meeting with Core Group and SEP in Paris the 19th November 2014.
1.0	08.12.14	Karima Bourquard		Quality Review

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

Glossary: **Definitions and Abbreviations**

Abbreviation	Explanation
Audit	Audit is an independent, objective assurance and consulting activity designed to add value and improve an organisation's operations. It helps an organisation accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes.
Business Plan	A Business Plan is a formal statement of a set of business goals, the reasons why they are believed attainable, and the plan for reaching those goals.
CAB	Conformance Assessment Body
Conformance testing	Testing the extent to which an Implementation Under Test is conforming to specific standards, guidelines or a specific profile.
HITCH	Healthcare Interoperability Testing and Conformance Harmonisation
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Standards Organisation
Process	A process is a set of activities that are interrelated or that interact with one another. Processes use resources to transform inputs into outputs. Processes are interconnected because the output from one process becomes the input for another process. In effect, processes are "glued" together by means of such input-output relationships.
Profile	A Profile is a selection of definitions and options from standards or other specifications. Profiling is usually done in order to achieve interoperability between different products and implementations since a profile aims at harmonizing all systems implementing it to use the same messages and contents. For example, IHE's so-called Integration Profiles selects messages and options from standards like HL7 or DICOM which are then implemented by all IHE-conformant systems. This ensures that IHE systems implementing the same Integration Profile are able to "talk the same language" in practice, thus enforcing interoperability between them.
Profiling	The process of creating a Profile (see Profile), used by "Profile definition".
QMS	See Quality Management System
Quality	The quality of something can be determined by comparing a set of inherent characteristics with a set of requirements. If those inherent characteristics meet all requirements, high or excellent quality is achieved. If those characteristics do not meet all requirements, a low or poor level of quality is achieved.
Quality Management System	A Quality Management System is a set of interrelated or interacting elements that organisations use to direct and control how quality policies are implemented and quality objectives are achieved. A process-based QMS uses a process approach to manage and control how its quality policy is implemented and quality objectives are achieved. A process-

	based QMS is a network of several interrelated and interconnected processes (elements). Each process uses resources to transform inputs into outputs. Since the output of one process becomes the input of another process, processes interact and are interrelated by means of such input-output relationships. These process interactions create a single process-based QMS.
Quality Manual	A Quality Manual documents an organisation's quality management system (QMS).
Quality Plan	A Quality Plan is a document that is used to specify the procedures and resources that are necessary to perform the processes... required reaching a set of quality objectives set as the final quality goals in the Quality Plan. Quality Plans also assign roles and persons to the corresponding tasks and specifies milestones to be reached within the scope of the quality plan. A Quality Plan is the result of Quality Planning.
Quality Planning	Quality Planning is the process of creating a Quality Plan (see Quality Plan).
Requirements to be tested	A condition or capability needed by a user to solve a problem or achieve an objective that must be met by a system or system component to satisfy a standard, specification, or other formally imposed document.
System Under Test (SUT)	A system in which the Implementation Under Test resides.
Test Execution	The process of running a test on the component or system under test, producing actual result(s).
Test Plan	A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.
Test Report	(Set of) document(s) that summarize(s) test results and other outcome information of a Test Execution. It also contains an evaluation of the corresponding test items against exit criteria.
Test Traces	Record of the logs that can be gathered by Systems under Test, Sniffers or other test tools during test execution. Logs can be gathered at different levels.
Tester	A skilled professional who is involved in the testing of a component or system.
Validation	Validation is a process. It uses objective evidence to confirm that the requirements which define an intended use or application have been met. Whenever all requirements have been met, a validated status is achieved. The process of validation can be carried out under realistic use conditions or within a simulated use environment.

Executive Summary

Today, it is a common requirement that eHealth solutions can share seamlessly data (i.e. are interoperable) between products from different vendors and across organisations. Unfortunately many solutions are not tested and implemented as specified and agreed before. This costs a lot of extra resources as many failures are only discovered once they are in daily operation. Unexpected failures leave customers and end-users with negative experience in using eHealth solutions in their daily practice and may seriously affect a patient's treatment and safety.

Implementing interoperability is complex and requires special attention to improve the quality of development as well as quality in use of the eHealth solutions. From a technical and interoperability perspective, quality is judged as if the system complies with agreed (international) requirements (eq. profiles and standards) and can exchange information with systems supporting the same standards.

The Quality Manual for Interoperability consists of two parts:

- Part I: Quality Management System (QMS) for Interoperability Testing
- Part II: Interoperability Testing Processes.

The Quality Manual is a customizable description and a set of templates with customization instructions that allow a Testing Entity to create its own, specific Interoperability Testing documentation in the form of a single Quality Manual for Interoperability Testing. End users and authorities may also use it confirming or recognising the quality and competencies of an Interoperability Testing Entity.

The key benefits for using a Quality Managements System for interoperability testing can be expressed in the three statements below:

- it will ensure continuous improvement of interoperability
- it will improve eHealth deployment
- it will facilitate the adoption of standards

Part II of the Quality Manual describes the Interoperability Testing Processes. The Interoperability Testing Processes is a set of interconnected "guidelines" that describes how to run a test session from start to end. Each process has defined input and output and can be maintained and improved in isolation and by different people with the required experience and skills.

The Interoperability Testing Processes are generic and can be used for IHE Connectathons, Continua Alliance Plugfests and regional Interoperability Testing, for example to test the implementation of ePrescription or Patient Summary.

The main processes for Interoperability Testing may include:

- Criteria and methods needed to ensure that both the operation and control of processes are effective

- Outline of necessary resources that are required in order to support the operation of the quality management system
- Monitoring, measurement (where applicable) and analysis of the processes in order to improve the quality of the interoperability testing.

The Interoperability Testing Processes are based on IEEE 829 and European best practice and includes:

1. Quality Planning
2. Test Plan Definition
3. Design Tests
4. Develop or Select Test Tools
5. Validation
6. Prepare Test Session
7. Test Plan Execution
8. Project Management
9. Process Management Update

Each of the nine interconnected Interoperability Testing Processes is described by using a generic template and a checklist on how each process can be adjusted to specific and/or local use.

The Quality Management System and requirements for the operation of Conformity Assessment Bodies (CAB) performing Interoperability Testing is describe in Part I of the Quality Manual (ANTILOPE WP2, D2.1 Quality Management System for Interoperability Testing).

1 Introduction

The Quality Manual for Interoperability Testing consists of two parts:

- Part I: A Quality Management System (QMS)
- Part II: A description of Interoperability Testing Processes (IT-P).

The Quality Manual is generic and includes detailed customization instructions that allow each organization on national or regional level to create their own, specific Interoperability Test documentation in form of a single manual.

The combination of including a QMS and quality procedures into a single manual is the preferred approach as it makes the documentation more user-friendly.

The Quality Manual will ensure uniform and transparent Interoperability Testing of eHealth systems across organizations and vendor's systems.

This document is part II of the Quality Manual for Interoperability Testing.

The Interoperability Testing Processes is a set of interconnected "guidelines" that describes how to run a test session from start to end. Each process has defined input and output and can be maintained and improved in isolation and by different people with the required experience and skills.

The Interoperability Testing Processes are generic and can be used for IHE Connectathons, Continua Alliance Plugfests and regional Interoperability Testing, for example to test the implementation of ePrescription or Patient Summary.

2 Actors and roles

This section provides a definition for the actors involved in the different interoperability QMS processes. The following table lists the minimum set of actors and roles that need to be defined in the interoperability quality manual.

Table 1: Definition of actors and their roles

Term	Definition
Top Level Management	The top level management coordinates the different activities. It gets reports from QA Manager, Test Manager and Auditors
QA Committee	A committee has the role to ensure the quality of the testing process, discusses the needs and decides on what needs to be done in terms of quality.
QA Manager	Manages the QA process. Gets input from the QA Committee and reports to Top Level Management.
Test Manager(TM)	Manages the testing. Organises the testing activities, reports to the Top Level Management. Follows the rules from the QA Committee to ensure the overall quality of the process
Testing team	Performs the tests and is under the supervision of the Test Manager.
SUT Operators	SUT Operators execute their SUTs test steps required by the test
Auditors	Auditors verify that the QMS process is correctly used. The auditor’s report to the Top Level Management.

In addition to the general actors and roles, some organisation may need to specify local actors and roles that are specific to the organisation. These are roles that do not always fit in the above list of roles but are implied by the testing strategy of the organisation. For instance, IHE defines a planning committee and a technical committee. Testing outcomes serve as an input to these committees to judge the appropriate and maturity of the provided specifications that in turn may be updated accordingly where needed.

→ Identify the persons and groups in your organisation that play the roles described above.

→ Define and describe the quality organisation.

3 Interoperability Testing Processes

Today, it is a common requirement that eHealth solutions can share data (i.e. are interoperable) seamlessly between products from different vendors and across organisations. Optimally these partners are involved in planning, either directly or via representatives.

Unfortunately many solutions are not tested and implemented as specified and agreed before. This costs a lot of extra resources as many failures are discovered as recently as when the solution is already in daily operation. The unexpected failure also leaves customers and end-users with negative experience in their daily work and may seriously affect a patient's treatment.

The testing organisation or group of organisations defines and agrees on the processes needed for interoperability in order to ensure that all customer and applicable legal requirements are met.

The main processes for interoperability QMS may include:

- Criteria and methods needed to ensure that both the operation and control of processes are effective
- Outline of necessary resources that are required in order to support the operation of the quality management system
- Monitoring, measurement (where applicable) and analysis of the processes in order to improve the quality of the interoperability testing.

Figure 1 introduces the testing process under the interoperability testing QMS. This process is based on IEEE 829.

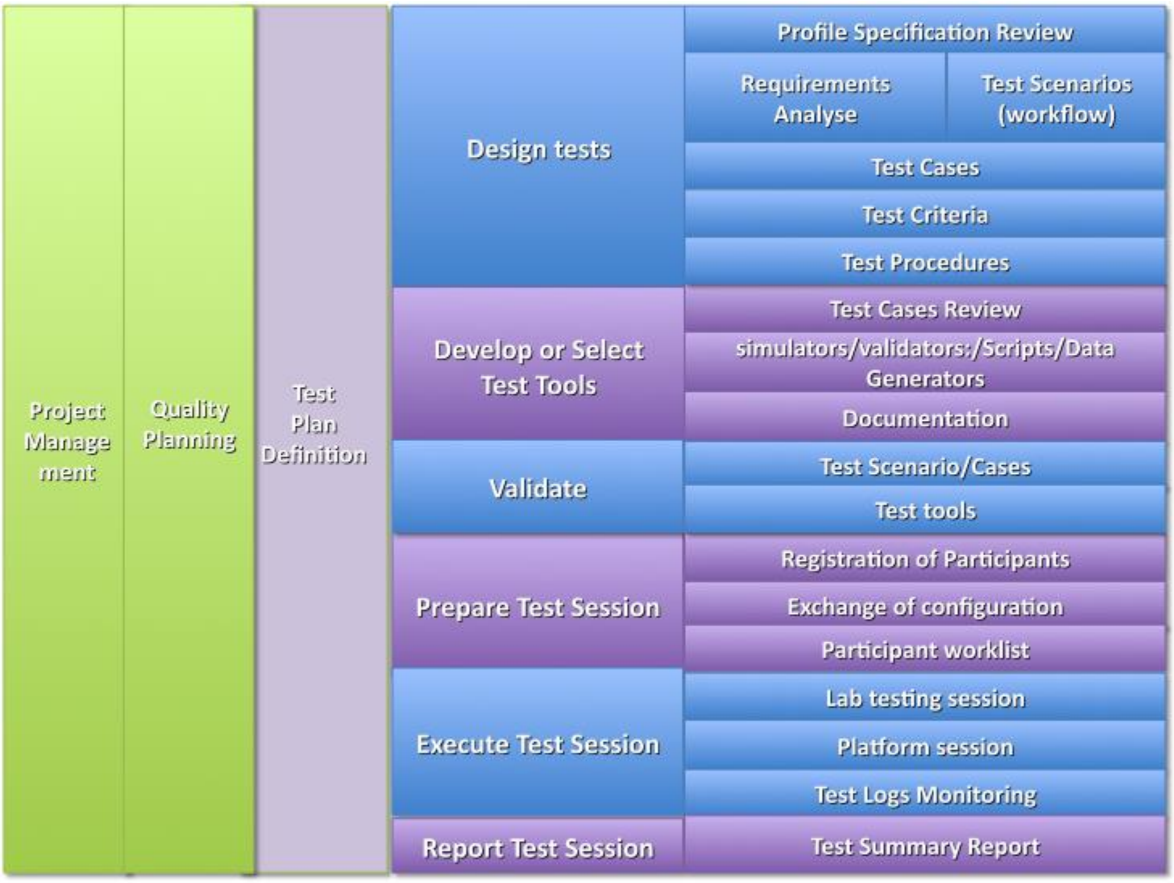

Figure 1: Testing Process and Activities

The organisation is committed to continually improve its effectiveness in meeting its objectives.

The interoperability QMS is made up of the processes for:

- Quality Planning (section 3.1)
- Test Plan Definition (section 3.2)
- Design Tests (section 3.3)
- Develop or Select Test Tools (section 3.4)
- Validation (section 3.6)
- Prepare Test Session (section 3.7)
- Test Plan Execution (section 6.7)
- Project Management (section 6.8)
- Process Management Update (section 6.9)

These processes can be refined on lower levels as shown in Figure 1. The listed processes are described in detail within the following sections.

→ List your own local interoperability QMS processes here, if needed, and describe them below.

3.1 PROCESS: Quality Planning

3.1.1 Why do we need planning?

Every project should have a quality plan but in reality very few actually define one.

It is evident that implementing interoperability is complex and requires special attention to improve the quality. From a technical and interoperability perspective, quality is judged as if the system complies with agreed (international) requirements (eq. profiles and standards) and can exchange information with systems supporting the same standards. Conformance and interoperability are not directly conditioning each other but mostly they go hand in hand.

There seem to be an overhead in undertaking quality management for interoperability but this is compensated by not having to fix failures further down the a product’s lifecycle. Inevitably, the later a problem is identified, the longer it usually takes fixing it.

→ Identify and describe why you need to plan the quality of interoperability testing. You can start backwards by identifying the existing main problems. Maybe you are not doing a systematic planning today?

3.1.2 Objective for planning

Quality planning is a systematic process that translates the quality policy and objectives into defined measurable requirements, and lays down a sequence of steps (processes) for realizing them within a specified timeframe.

The requirements from the quality planning must be defined SMART:

- Specific
- Measurable
- Attainable
- Relevant
- Time-bound

→ Describe the main requirements for your interoperability testing. Use the SMART qualifiers when defining the objectives.

3.1.3 Producing a quality plan

Producing a quality plan for interoperability testing is not specifically complex compared to all other required planning activities in product development. It involves identifying all the deliverables and deciding how to best validate their quality.

A quality plan for interoperability testing needs to cover a number of elements:

- What needs to go through a quality check?
- What is the most appropriate way to check the quality?
- When should it be carried out?
- What materials should be used for the quality check?

These issues are described in more detail below.

3.1.3.1 Background material

The underlying basis for the quality plan for interoperability testing is:

- The organisation's strategy
- The organisation's business plan

→ Identify the background material for preparing your quality plan. If these documents are not yet defined and agreed in your organisation, this work must be done before proceeding with the quality planning.

→ Based on the background material, prepare a list of deliverables, which are used for interoperability testing (eq. list of test tools, list of standards, test scenarios).

3.1.3.2 What needs to go through a quality check?

Typically what needs to be checked are the deliverables, which are used for the interoperability testing. Any significant deliverable should have some form of quality check carried out.

For example, a profile document should be considered significant and shall go through a quality check.

A test tool selected for testing a product should most of the times be considered significant and therefore shall go through a quality check.

A memo or weekly report may not be significant and will not go through a quality check.

→ Update the list of deliverable and add codes and priority, in order to ensure that the most important deliverables will go through a quality check.

3.1.3.3 What is the most appropriate way to check the quality?

The most appropriate way to check the quality of a deliverable is thinking backwards. If the end result is that a particular deliverable should be used to test how good an application meets a standard, then part of the quality checking should focus on compliance with the standard. This would indicate a "Standard review" of the deliverable could be the best approach.

If the deliverable is a tool for testing exchange of data between systems, the final part of the quality checking could differentiate between "beta" or "final and stable".

→ Update the list of deliverables and add means how their quality should be checked. The work can start be defining the different ways you want to check the quality and how to allocate needed and qualified resources.

3.1.3.4 When should it be carried out?

Most "Quality Events" are held just prior to the completion of the deliverable. If however there are long development lead times for a deliverable, it might be sensible to hold (for example) earlier "Quality Events".

For example, if development of tools for a particular test will take 10 weeks, it may be worth holding a code inspection after four weeks to identify any problems and initiate corrective measures.

→ Update the list of deliverables and add resources and milestones for each. For important and high risk deliverables, eq. software development, intermediate quality checks can be planned.

3.1.3.5 What materials should be used for the quality check?

The deliverables will usually be self-evident, but it may be useful to reduce things to checklists in order to make them more manageable and ensure that the most important areas are checked, rather than the full deliverable.

→ Provide a clear description of the main areas to be checked in a specific deliverable. It must not be left to the reviewers to decide what to check, as they will most probably check the areas where they have the best competences. The areas to be checked are the ones where it is discussed and agreed that there is potential or need for improving the interoperability testing.

3.1.4 **Risk Planning**

Risk Planning helps to foresee risks and identifies actions to prevent them. Also it reduces the risks' impact if they actually emerge.

The Risk Plan is created as part of the quality planning process. It lists all foreseeable risks, their ranking and priority, the preventive and contingent actions, along with a process for tracking them. A Risk Plan template will help you perform these steps quickly and easily.

A Risk Plan should be used whenever risks need to be carefully managed. For instance, during the start-up of a project, a Risk Plan is created to identify and manage the risk involved with the project delivery. The Risk Plan is referred to frequently throughout the project, to ensure that all risks are mitigated as quickly as possible. A Risk Plan template should minimally include:

- Listing of likely risks, which will jeopardise the interoperability testing
- Identification of preventative actions to prevent the risk from occurring
- Listing of contingent actions to reduce the impact, should the risk occur
- Schedule of these actions within an acceptable timeframe
- Rules for monitoring the status of each risk throughout the project

The first step in creating a Risk Plan is to identify the likely risks which may affect the project. A series of risk categories is identified and for each category a suite of potential risks is listed. This may take place during a Risk Planning workshop, involving each of the key project stakeholders who are involved in or affected by the project. This may include the project manager, team, suppliers, and/or the end user. Each of the risks identified is described in detail and documented within the Risk Plan. Examples of Risk are too few educated Testers, significant errors in a profile and unstable Test Tool.

→ Prepare a Risk Plan for the entire interoperability testing. Start the work by preparing a Risk Plan template to be used as a part of the quality planning.

3.1.5 Planning roles and responsibilities

The QA manager is responsible for preparing the quality plan.

The quality plan should be approved by the top level management to ensure high visibility and to ensure that it is in alignment with the core business and strategy of the organisation.

→ As a QA manager, prepare the quality plan including the risk plan and ask the top level management for approval.

3.2 PROCESS: Test Plan Definition

3.2.1 Why do we need a test plan definition?

Test plan definitions are needed because the interoperability testing is a complex activity and can be clearly identified as a project on its own, with several tasks. This project needs people with testing and development skills as well as managers organising and monitoring the testing processes.

The test plan will help the testers to be sure that conformance and interoperability requirements defined in the applying specifications are reasonably tested according the risk assessments that was defined within the test strategy.

3.2.2 Objective for test plan definition

The test plan definition will describe the test strategy and its implementation: all activities are carefully defined and planned in order to test profile specification in a given context.

3.2.3 Work to be done for the test plan definition

The test plan identifies all the activities to be detailed planned.

To build the test plan, two aspects should be considered:

- The extent of the testing in relation to achieve the level of acceptance to the conformance of the test cases;
- The risk assessment: it will help to define a testing plan considering the quality and the complexity of the specifications, the number of the tests and their organisation, the relationships between test conditions and the test procedures.

Furthermore, the following questions should be answered:

- What is required to be tested in order to make sure that the interoperability objectives are met ;
- How to test it and how to verify that is was correctly tested.

To achieve the objective of the conformance and interoperability testing, several tasks have to be performed. The goal, the input and the deliverables of each task will be described in the test plan definition according to the level of expected quality and the scope of the interoperability testing process.

The activities for assembling a test plan consist of:

- Definition of the scope and objective: the list of features to be tested as well as the features that are explicitly not tested must be described; criteria to accept the system (test acceptance), etc.;
- Specification of the test design: the list of test scenarios, test cases, requirements according to the profile specifications;
- Development or the selection of the test tools: first approach on the test tools needs (type of tools, data generators, etc.);
- Preparation of the test session: environment, configuration, test platform, etc.;
- Execution of the test session: session type and session numbers, schedule, training, etc.;
- Reporting of test results.

For each activity, the scope, requirements, persons and material resources, responsibilities of the actors participating in the conformance and interoperability testing process, risk assessments and schedule information are documented.

Also, existing methodologies for test plan definitions like IEEE 829 should be assessed on a regular basis in order to develop state-of-the art test plans.

→ Start by identifying existing background material for preparing a test plan (e.g. a list of existing documents describing test cases, test environment and test tools).

→ Define the test plan for your organisation.

3.2.4 Test plan definition risks and failures

The equilibrium between resources, schedule and the test design needs to be established. If the scope of testing is chosen too large or narrow this may be a cause of failure. If the test plan is too ambitious and needs too much effort, the full execution of the test plan will be unrealistic.

A bad risk assessment and a weakness on the requirements specifications are also possible causes of failure. This is generally due to a lack of knowledge of what needs to be tested or a lack of testing skills and experience.

If customers notice a weakness in the quality of products, they will no longer have confidence in the testing process of a particular project.

If the feedback to the organisation that has made the specification (eq. a profile for patient identification) is not well documented, the testing process has no sense or will be the bad quality.

3.2.5 Test plan roles and responsibilities

The test plan definition is under the responsibility of the Test Manager who organises the complete testing process.

3.3 PROCESS: Design Tests

3.3.1 Why do we need a Test Design?

The design test process is needed to produce detailed tests that correspond to the interoperability scope objectives. These tests aim at validating the information that is exchanged between eHealth systems or other products, contributing to the end-users to trust in interoperability of such systems.

3.3.2 Objective for Test Design

The objective of the Test Design is to produce test cases that will test the requirements and define acceptance criteria that will be used during the test execution session.

3.3.3 Work to be done

The activities to realise the test specifications are:

- Review the profile specifications: The profile specifications are reviewed deeply and each relevant requirement/reference is picked up. That is needed to select and design the correct tests. At the same occasion, the consistency of the specification will be detected and documented during that review with the aim establish and define in a feedback process with the profile specification team.
- Analyse the requirements: the list of the System Under Tests (SUTs) conformance and interoperability requirements is analysed and leads to the definition of the test cases (which role the SUT has to play, the transactions/messages/options/format and behaviour to be implemented). During this task, new requirements may show up and then be taken over.
- Define the test scenario (workflow): a business case is described in terms of workflow and interaction between systems (sometimes also called actors) as explained in the profile specification. The message triggers (i.e. events), the messages itself and the expected response messages and codes returned as well as the planned and possible sequencing of message are identified.
- Define test cases: For one business case, several test cases including steps can be described corresponding to the behaviours of the SUT and its roles. The test data is defined at the same time (see next bullet point). The detailed test case specification should be created in an iterative approach that allows the test team to better understand the testing context and to increase the quality of the test specification.
- Define the test data to be used during the test execution.
- Define acceptance criteria (criteria indicating that the test has been passed or failed): For each test scenario or test case, the test steps to be passed together with their expected

results and the criteria are listed.

- Define test procedures: specify how to execute the test, the inputs, the steps, the conditions to be met and the expected outputs (logs, traces and expected results).

The definition of test scenarios and test cases is supported by a methodology such as:

- Analytic methodology based on deeper analysis of the underlying profile to be tested and the risk analysis performed. Both lead to the definition of test cases;
- Heuristic methodology based on selection of previous reported errors that should be tested for and that should corrected in the future.

The test cases are described in detail with the help of profile analysts and should be developed in an iterative approach that allows the testers to better understand the context and finally to improve the quality of the specifications.

Figure 4 shows an example of test case with its steps, taken from the IHE ITI Technical Framework.

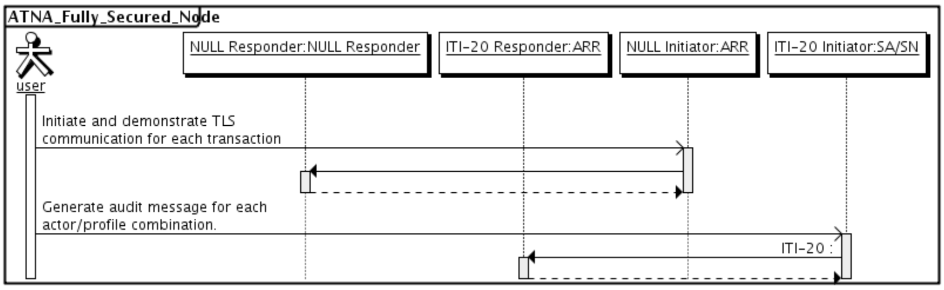
ATNA_Fully_Secured_Node
user
NULL Responder:NULL Responder
ITI-20 Responder:ARR
NULL Initiator:ARR
ITI-20 Initiator:SA/SN
Initiate and demonstrate TLS communication for each transaction
Generate audit message for each actor/profile combination.
ITI-20 :

Figure 2: Test steps defined for ATNA profile corresponding to the test case ”Fully Secured Node” documented and registered in IHE Gazelle tool

The outputs of the test design process are:

- The detailed specifications of the tests (test scenarios and test cases) including conformance tests;
- The test phases and procedures including the test configuration and environment management;
- The test and configuration data specifications.

→ Define the tests design specification and write all the detailed documents for each output (detailed specification document, test phases and procedures document, test and configuration data document)

3.3.4 Risks and failures in Test Design

Risks arise from a possible lack of skills and experience of the team (background on standards, profiles, testing environment) and from a limited availability of profile analysts who can support the development of the test cases.

Failures may be caused by:

- A scope too large or not clear;
- A risk assessment not sufficiently analysed with the goal;
- Too many test cases or test scenarios which require too much effort;
- The expected results are not in line with the expected quality level or the expected quality is not reachable at all.

3.3.5 Design tests roles and responsibilities

The Test Manager has the responsibility of the overall tasks together with the testing team, which has the responsibility to write test cases. The latter is supported by the profile specification team.

3.4 PROCESS: Develop or Select Test Tools

This process is about the selection of test tools needed to run the tests. Existing tools shall be reviewed and considered for selection as developing new tools is time consuming. Utilizing or extending existing tools should be preferred when the choice is available.

Three categories of tools have been identified:

- Tools for test management and reporting;
- Tools for conformance testing;
- Tools for interoperability testing.

However, other categorisations may be possible and the borders between the listed categories can often not be set clearly.

3.4.1 Why do we need test tools

The main advantage of using test tools is to automate the task of testing and often, to speed up the testing process and therefore testing efficiency. They allow repeating tests as consistently as possible and as such contribute to a constant level of quality. Furthermore, using existing test tools makes sure to profit from the experiences of others.

Sheer size of the test space is another argument for tool-based testing. Interoperability indeed requires the exchange of multiple messages between devices (SUTs and simulators that simulate connection partners for testing). The variety of processes also adds to the typical number of test cases. The number of messages that a typical test includes is therefore usually far beyond what is feasible in manual testing.

Test tools from different categories are needed, depending on the objective of testing and the test design. A global architecture must be defined (and maintained) around all test tools and must cover different aspects (eq. configuration, version, libraries and documentation).

The benefits of the usage of test tools are multiple:

- To reduce the effort necessary for tests that need to be repeated frequently;
- To increase the consistency of the tests;
- To evaluate with objectivity (the tests are repeatable and therefore often more reliable than the tests made by a human observer);
- To trace and to provide access to results of tests that were already executed;
- To assure that the same tests are executed for every systems as needed;
- To share the test scenarios and tools among several groups and projects.

3.4.2 Objective of selection and development of the test tools

Test tools are used in order to test interaction between two or more systems.

The main categories widely used in eHealth conformance and interoperability testing environments are the following:

- Tools for test management and reporting: ”These tools are able to manage the tests (automatic or not) and often include requirements and/or specifications modules that allow to automatically generate the RTM (Requirement Test Matrix) which is one of the main metric to know the functional coverage of the SUT (System Under Test)”. These tools should guarantee the traceability of all the tests, first and foremost by producing test reports. In the context of interoperability testing, test management tools also may include functionality to identify possible test partners from a pool of actual systems under test and/or simulators for specific tests. This is especially important if cross-vendor testing is performed, i.e. systems from multiple vendors are connected for testing at a testing event like the Connectathons performed by IHE or “virtual” test sessions, e.g. over the Internet.

→ Define how test management tools in your test scenario can support test partners, test tools and test runs. Consider using existing tools where possible. However, for small scenarios, be sure to not “oversize” your test architecture, i.e. start managed and small but stay extensible. Note that in your organisation you might be more specific than it is shown here.

- Test tools dedicated to conformance checking: these tools are used to test the conformance of the messages sent by the SUT regarding the profile specifications (standards, etc.). Tools that do this kind of checking are usually called validators.
- Test tools dedicated to interoperability checking: In contradiction to conformance checking tools, these tools are utilised in order to ensure that two or more systems actually are able to exchange data with each other and understand the data exchanged. Thus, they might but do not necessarily validate that the data exchange is conformant to the underlying specifications, i.e. a test can be successful if both systems “understand” each other even if they violate rules of the specifications. Simulators might support the interoperability testing of systems by mimicking test partners. Often, many tools involve an adequate mixture of interoperability and conformance testing and so does the full test process.

In each category, different types of test tools can be distinguished and actually used to test for different aspects of a SUT. The techniques to specify test scenarios and test tools are described in the section below.

Some tools may be specific to the SUT type tested and may not yet exist. Thus, they have to be specified and developed for the organisation. Some tools may already be available on the market and may be sufficient to fulfil the test requirements. One difficulty in the task will be to identify the tools that need to be developed specifically for a specific series of tests and the tools that can be reused in other contexts.

3.4.3 **Work to be done**

The test tool development heavily depends on the tests case definitions. Thus, the first task to be done is to review the test cases. Requirements need to be understood in order to perform testing successfully.

The following types of tests can be distinguished in order to specify tests and evaluate their results:

- Checking whether data meets the expected values;
- Checking the SUT's behaviour if data value is out of the valid value range in order to check the robustness of the SUT;
- Analysis by decision table (conditions and actions are identified): the test will check the condition branches possible, i.e. different messaging orders and filled-in values possible are triggered and checked;
- Analysis of a global test scenario: Test scenarios registered in test management tools and directly executed and analysed by a human observer or automatically by test tools, analysis of return codes and global test results. This is kind of an encompassing type of test.

Test tools for test management can be classified according to the following table (not exhaustive):

Table 2 – Categories of testing tools

Tools	Role	Functionalities
Tool managing and supporting the test	Test management	Manage the tests between SUTs Registration of the SUts and their profile Configuration of the test environment Integrate test tools Sample sharing Test execution Traceability of the tests Test measurement Reporting results Statistics
	Management of the requirements	The tests cases and requirements are

Tools	Role	Functionalities
		linked
	Management of the incidents	Report the errors, default of the tools and the resolution of the incidents
	Configuration management	Configure the test environment
Tool used for the designing tests	Specification of tests and test scenarios	Interfaces (e.g. GUI) used for the creation and management of test scenarios
	Specification of test data	Interfaces(e.g.) GUI for the creation and management of the test data
Tool used for the execution of the test	Scripts	The tests are executed automatically or semi automatically and are repeatable. The execution of the tests can be registered and stored with the entry data and the results
	Simulators	This tool is used to simulate an actor or system
	Validators	Tools that check the conformity of data or messages generated by SUTs
	Tools for security environment	Tools used for testing security test cases, for example the generation of test certificates
	Tools to generate errors and defaults	
	Tools to generate reference data	tools generating test data or samples for the test
Tools for performance testing	Tools for stress testing	

Each tool must be tested and documented. It should be integrated in a testing platform such as a test tool management system. Inside such a platform, automated test scenarios will be well specified and generally optimised to be more efficient than a test manually executed by a human observer. In some cases, this platform might be available and open for multiple companies, institutions or any other party that likes to get involved. Every participating organisation will be able to test its own systems and to define its specific testing environments, configurations in a common manner. Such a test platform should offer the following functionalities in order to assure quality:

- Documentation for deploying the tool;

- Clearly defined configuration mechanisms;
- Openness to facilitate the integration with other tools, e. g. to control and trigger simulators or validators;
- Easy to execute and also replay a test if applicable;
- Reporting, logging and traceability of test setup and test runs;
- Large community using the tool if possible, providing a better chance that tool is proven to work and that testing with other community members is facilitated.

→ Define functional requirements

→ Define the test documentation requirements

→ Define acceptance criteria

In order to ensure the quality of tool development, the following is recommendable:

- Use of a Source Code Management tool to manage revisions of the source code of tools if applicable;
- Provision of user and developer documentation of the tools;
- Testing of the tools itself and establishment of a procedure for deploying new versions into the test cycle;
- Set up a bug tracking system.

→ Define the test tool development requirements

→ Define bug tracking system requirements

→ Define continuous integration of testing components

→ Define distribution of tools component

→ Define versioning of the components

→ Define developer and user documentation

3.4.4 **Risks and Failures**

The major risk of using (a wrong, insufficient or buggy) test tools is to produce results that do not resemble the actual performance of the tested units. There are various reasons that may lead to wrong results, e.g.:

- The test scenarios are not well defined ;
- The tool has bugs which are not known before testing, or even worse, are not recognized even while or after testing;
- The test data is incorrect or incomplete;
- The test reports produced are incorrect or incomplete;
- The tool is used incorrectly due to lack of training.

Weaknesses of the test tools and incorrect usage have severe impact on the performance of testing and later in practice (e.g. no access to the valid eHealth data, wrong diagnosis). This will influence the trust that end users have in the tested systems and in the testing authorities (if the testing is performed by a third party, e.g. in the context of a quality assessment or certification) as well. A test tool with bugs or other insufficiencies will lose the confidence of the testing team and will (and must) finally be abandoned.

A risk assessment must be applied when a testing environment is set up. It will cover the following topics (not exhaustive):

- Are the specifications of the test scenarios available in order to perform adequate testing of the test tool itself?
- Has the test tool passed validation test with success?
- Are known bugs identified and guaranteed to be handled correctly within the test runs?
- Has this tool already been used within other test platforms and did it work satisfactorily?
- Is the tool's community still active and supportive?
- Does the tool match the requirements to be tested?

A risk assessment will include the following steps:

1. Identification of the risks;
2. Analysis of the risks;
3. Solution to decrease the risks;
4. Identification of the residual risks.

People involved into this analysis are testers (e.g. engineers) involved in the profile specification and tool developers.

A process of continuous improvement must be in place and will help to decrease the number of flaws in the tool and therefore increase its quality.

→ Develop risks assessment document for developing, selecting and configuring test tools

3.4.5 Roles and responsibility

The most important requirement is that the team who develops the test tools (test tool developer team) must be different from the team who has developed the profile specification (profile specification team).

However, both teams must not be completely isolated from each other. Particularly for test management tools, there also needs communication with the developers of those systems that are going to be tested with the tool. Since both are going to link together in order to perform a set of specific interoperability tests, it must be assured that the test management tool offers adequate interfaces.

The test tool developer team also needs skills regarding the testing process and on all development environments that are needed.

The testing team has the responsibility to review clear test plans, which will be communicated to the candidate system developer teams sufficiently in advance. The test tool developer team and the testing team could be the same or two different teams, the first has the responsibility of developing test tools, the second to organise the testing, i.e. organise the testing platform, execute the tests as well as report and publishing the results. The two teams must be motivated and should follow the test plan very carefully. Often, for bigger interoperability testing initiatives like IHE, the auditor team is at least partially different from the development team since to the large scope, many tools are not written from scratch but instead are reused, e. g. as provided from the Open Source community. In those cases, the auditor mainly selects and configures existing tools and makes sure the quality of each tool and their combination permits adequate testing.

The test tool developer team should follow the quality assurance set up for the development of the test tools and should develop the test tools taking into account the state of art in this field.

The auditor team should precisely register the test scenarios, configuration and all information needed for the execution of the tests and all the results of the test in the testing tool management.

3.5 PROCESS: Validation

3.5.1 Why do we need validation?

Testing is essential in achieving interoperability of eHealth systems. As testing is based on test cases and may involve test tools, their quality has a very big impact on the testing process itself and its results. The quality of the test cases and tools must be validated before they can be deployed in serious testing. Unreliable test tools may result in loss of time and effort in testing, potentially leading to a serious loss of confidence in the whole process that is originally aiming at enhancing interoperability.

It is important to note that test cases and test tools are developed on the basis of relevant standards and profiles in the same way as the eHealth systems they should actually test. During validation, test cases and test tools are used for the first time to test eHealth systems. Actual test runs often reveal issues that were not expected and that may require improvements in test cases or test tools. Also, as standards are never perfect, validation can also reveal problems in relevant standards, profiles and other specifications.

3.5.2 **Objective for validation**

The objective of validation is to ensure and, to the extent needed, enhance the quality of test cases and test tools to a level that is found appropriate for using them for testing eHealth systems.

To give a slightly more comprehensive definition of validation, a definition used for validation of standards is reused in a simplified form.

"Validation is the process, with associated methods, procedures and tools, by which an evaluation is made that a test plan can be fully implemented, conforms to rules for test plans and satisfies the purpose expressed in the record of requirements on which the test plan is based.”

Validation activity needs to be planned and well-coordinated between various actors that will be involved.

The inputs of validation activities are standards, profiles and other specifications as well as test cases and test tools that are to be used for testing the interoperability of eHealth applications and products.

The primary output of the validation activity are test cases and test tools that are proven to be fit for use in testing interoperability of eHealth systems. During validation interim problem reports are generated and used to drive the improvement of test cases and tools. The validation process is closed with a validation report. Additional output of the validation activity may also be requests to improve relevant standards and/or profiles as they may be found to be ambiguous or even contain contradictory requirements.

Two validation processes can be identified:

- Test Case Validation
- Test Tool Validation

The level of quality is an engineering compromise between the cost of and time required for validation and fixing the eHealth systems correspondingly on one side and estimated impact of interoperability problems on the other side. The bigger the consequences of interoperability failures could be, the more investment in validation is justified.

A clear objective of test tool validation is that all test tools that could be used for a given profile always give same test results if performed with the same, unchanged system under test.

3.5.3 Work to be done

The test cases can be validated through peer review and by actually running them.

In peer review, the test cases are checked against relevant standards and profiles as well as their intended use. Peer review could consist of following checks:

- Check that all relevant standards and other specifications are correctly referenced.
- Check that all statements in test cases are unambiguous as well as clear and precise.
- Check that all requirements are well covered by test cases.
- Check that test cases do not test a requirement that cannot be traced back to a standard or profile.
- Check that the test cases could actually be used in testing (to the extent possible).
- Check that test messages and their content are well defined.
- Check that expected message sequencing is well defined.
- Check that optionality of message content and message sequencing is well defined.
- Check that the test verdict criteria (passed versus failed) are well defined.

As the term peer implies, in order to spot potential problems the reviewer needs to be different from the person that developed the test cases.

While the above could greatly improve the test cases, ultimately they need to be validated by using them in a pre-run for testing a suitable number of real eHealth systems.

During validation pre-runs, the following needs to be done:

- Each test needs to be run against a number of eHealth systems implementing a given profile
- Traces of each test run need to be collected
- Experienced test experts need to examine the recorded traces and confirm that the trace is as specified for a test case, checking all aspects that may be relevant (messages, message content (presence/absence, values), timing, and other conditions.
- Any problem found during this examination needs to be recorded and reported to those that could fix it (system under test developers, test case developers or standard/profile developers)
- Once problems are fixed, the failed test case is run again and new traces are examined.

Since test tools bring a degree of automation into interoperability testing, it is even more important that they work reliably well. While peer code and documentation review does help, nothing can replace actual test runs with detailed examination of test traces.

During test tool validation the primary activity is to run and examine test cases on a particular tool. In addition, there are other capabilities of test tools that need to be checked:

- Ability to work over required types of networks

- Ability to run over all underlying protocols allowed by the profile
- Ability to be configured as required (addresses parameters etc.)
- Ability to record traces, perhaps in a particular format
- Ability to display and record log statements
- Ability to be integrated with test management tools
- Tools must be fast enough, if relevant
- Further requirements specific to the standards, profiles and type of SUT tested

Organisations that validate test tools need to define in their validation quality plans describing how they intend to organise the validation process. As an example, one validation program requires that two different test tools testing two different SUTs pass the tests with traces that satisfy the examiners during validation. Another similar validation programme requires every tool to be validated against three different SUTs.

→ Define a plan how the test case validation will be performed.

→ Define a plan how the test tool validation will be performed.

Some guidance in developing the above plans can be found in [5], in particular in preparing a choice of validation methods, in estimating the resource requirements and planning the time required.

During validation, various data need to be collected and communicated. Web-based tools with predefined entry and report generation means are particularly suited for this. For any additional reports that need to flow between various roles, templates should be defined in advance.

→ Define the validation data collection means

→ Define communication flows between roles during validation

→ Define templates or web-based tools for various reports

Validity status of test cases and test tools will keep changing over time. During their validation, status changes may be rather frequent. Later when test cases and test tools are considered as validated and used for testing, their validity status should change less frequently but will nevertheless undergo changes as additional test cases/tools get validated or as problems are found resulting in a test case or test tool to be (temporarily) downgraded. The status of test plans and test tools needs therefore to be tracked. While manual tracking is possible in simple cases, using tracking tools is more productive. For many reasons, one of the best ways today is to use databases that are accessible via a web-frontend. This is equally valid for problem report entry, status updates, planning of required corrective activities and last but not least monitoring the status changes.

→ Define means of tracking validity status of test cases and test tools.

→ Define various statuses that a test case/test tool may have and the rules applying to status transitions.

3.5.4 Validation risks and failures

Validation requires a high degree of coordination between several partners. As test cases need to be executed against a number of eHealth systems, everything needs to be available at the required point in time. Moreover, as problems are detected, all relevant entities need to make required corrections available in a timely manner. The test case developers may need to update the test case description or code that implements the test, eHealth systems developers may need to update their implementation, other parties defining standards and/or profiles may need to improve their documents. There is a strong risk that delayed reactions of any entity may lead to serious delays or, even worse, absence of reaction may lead to an inability of validating test cases.

Validation requires skilled human resources that:

- have a testing background and training with, in particular, strong ability to spot potential problems during trace examination,
- have personal inclination and ability to insist on details where this could be important for testing,
- understand relevant standards/profiles,
- are not involved in the development of the very standard or profile they validate
- have the ability to manage reported problems as well as tracking down problem solutions and
- have the ability to effectively communicate with various partners during phases of problem identification and resolution.

If not enough human resources with adequate skills are considered in the planning, there is a high risk that validation is delayed or not fully achieved. It should be understood that delays in completing validation could be very damaging, as the time window when the results are needed depends on the deployment schedule of systems implementing a given profile. Once systems are widely deployed, it becomes more difficult to make them interoperable.

Even with very good validation, some problems may remain undetected. Testing may proceed without problems until some system under test fails the tests but manages to demonstrate that it is in fact behaving correctly and that the test case or test tool need to be improved. It is crucial that the number of such situations remains small, and therefore does not disturb the confidence in the validation results. Any loss of confidence in validations results could have a very negative impact on the whole interoperability enhancement program.

One of the big risk factors for interoperability is the stability of the standards and profiles. Testing for interoperability is performed over longer periods of time and during both validation and testing a stable basis is required.

Standards and profiles will evolve over time and one version or release will be followed by new revisions. Test case specification will have to evolve accordingly as well and test tools will have to be upgraded. Each time this happens, partial revalidation of test cases and test tools will be required.

Also, it is very probable that those new and old profile implementations will have to coexist and that new test cases may be required.

3.5.5 Validation roles and responsibilities

Each organisation should define its own way of performing validation. An example of specific validation roles and responsibilities could be:

- Peer reviewers
- Testers that perform test runs and collect traces
- SUT operators required to initiate required actions of their system during testing, for example entering example data on the user interface and triggering a transaction
- Persons that examine traces and report successful test runs for the purpose of validating the test cases/test tools
- Examine failed runs with the intention of resolving problems found in test cases and tools
- Persons that improve those parts of the test tools that were identified as being problematic (hardware, software)
- Persons that improve test case specifications
- A body (committee) that decides:
- What is acceptable and what is not (test cases, tools and test results)
- Which test plans and tools are ready and at which point in time they should be in testing

→ Define roles specific to your organisation

→ Define and document the rules guiding the validation decisions

3.6 PROCESS: Prepare Test Session

3.6.1 Why do we need to prepare a test session?

A test session can either take the form of a face-to-face meeting or a virtual meeting over the Internet. Sometimes, interoperability testing is also done in-house if the communication partners (e.g. other products, but commonly free implementations, simulators or other test tools) are available. The latter method is sometimes also referred to as lab testing. One could also think of mixed approaches, where a test session includes some in-house testing and then moves on to virtual testing. Either form needs to be prepared in order to be successful.

3.6.2 Objective for the preparation of a test session

The objective of preparing a test session with all participants is mainly to reduce the risks associated with test sessions failing or not being able to start at all, thus putting unnecessary workload and time on the participants.

3.6.3 Work to be done

The work to be done basically concerns the training of test participants and the preparation of the organisation responsible for test session / test event logistics. Also, participants must prepare to the test sessions by preparing their systems, i.e. they must configure other communication partners (network addresses, coded terminology and so on).

→ Define which model of interoperability testing fits best for your organisation (face-to-face, in-house, virtual or mixed). If required, you may also define a strategy for the longer term, e.g. start with in-house testing, and then advance to virtual testing methods and finally visit face-to-face interoperability testing meetings according to a pre-defined time schedule.

→ Training material for SUT operators

→ Training material for auditors who are responsible for rating test runs as being successful or failed. Sometimes there might not be an institution or human playing the role of an auditor at all, but for virtual and in-house testing there could be software, e. g. as part of the test management system) doing all the audits. On the long term, of course, neutral human auditors (supported by software) should be put in favour.

→ Define the procedure for testing team recruitment, if applicable. Especially for third parties organizing test events, it may turn out difficult to find neutral, well-trained experts that do not belong to one of the organisations that are under test.

→ Define rules for participation: Define which systems are about (or are required) to join the test session and which pre-requisites are to be fulfilled for that. For example, SUTs do have to provide the documentation before or need to show evidence (e. g. created from testing software) that they provide basic, working functionality that is required for the test session.

→ Logistics requirement of test platform (power, network, catering, security…). This is especially needed if face-to-face test events are organised. There are many requirements associated with these logistics that are, however, beyond the scope of this document.

→ Organise registration and communication. Participants need to register in advance and exchange configuration parameters with partners. Failure to share configurations and to pre-load them into the systems under test can result in a waste of time during the face-to-face test events.

3.6.4 Test session preparation risks and failures

The following risks are bound to the test session preparation:

- Insufficient or inadequate training material for participants (SUT operators and auditors);
- Bad estimation of the requirements (e. g. computational power, disk space, network bandwidth) for the testing platform. An underestimation may result in difficulties during the testing session. An overestimation may jeopardise the budget of the testing session;
- Bad communication and advertising resulting in participants not registering to the test session;

- If the number of auditors recruited to verify the test is inadequate, this may results in a large backlog of test to verify. Auditors get under pressure and it may decrease the quality of their verification of the tests.

→ Note or check that the risk are properly addressed in the test management

3.6.5 Test session preparation roles and responsibilities

The preparation is performed by the test manager under the supervision of the top level management. The test manager is in contact with the SUT operators during all the test session preparation phase. He/she acts as the moderator, ensuring that all the parties concerned coordinate and that planned milestones are respected. The test manager is responsible to report to the top-level management of any issues that could jeopardise the organisation of a test session.

→ List the roles and responsibilities of the test manager in the preparation of the test session.

3.7 PROCESS: Test Plan Execution

Six processes for Test Plan Execution have been identified:

- Identify system under test (SUT) and testers;
- List test plan and SUTs;
- Recruit and train testing team;
- Test execution;
- Test report;
- Test result validation.

3.7.1 Why do we need a test plan execution?

Test plan execution is the phase where tests are actually performed.

3.7.2 Objective for test plan execution

Test plan execution might include lab testing and platform testing. Here, with lab testing the in-house pre-testing is meant that takes place before actually participating in a larger test event (e.g. face-to-face) that is (most probably) supported by test management tool (platform testing).

Lab testing is usually an in-house testing activity that aims at the acceptance criteria and preparation of the SUTs for the platform testing session. The objective of the lab testing is to gather enough evidence that a SUT is well-prepared for the platform testing, i.e. has chances to pass many tests in the platform testing session. Participants not being adequately prepared also cost valuable time and work of other session participants that perform interoperability tests with that system.

Platform testing time is usually quite limited, especially in face-to-face meetings but also in virtual test setups, because participants must work together and therefore negotiate common time slots. Thus, optimally every minute of the platform testing session should be utilized for actual testing. Overall, test plan execution needs to be well planned and prepared for participants to gain most benefit for all involved parties.

3.7.3 Work to be done

Test plan execution can be successful if the following tasks are completed:

- Define key performance indicators (KPI) that allow following the progress of the test execution: number of test performed, SUT not testing, SUT currently testing with set of other systems, configuration of the SUT…
- Define reporting elements that need to be provided in the test session report

→ Define relevant KPI for your testing session

→ Define reporting elements

→ Define satisfaction questionnaire for participants (SUT operators and testing team)

3.7.4 Test plan execution risks and failures

The risks bound to test executions are the followings:

- Missing partners: some test cases may not be tested due to lack of participants. Sometimes a simulator might be used as a (poor) replacement;
- SUT has too many tests to perform within the timeframe foreseen for the test session;
- Defective infrastructure: power supply or networking problems as well as failure of the test management system or other central technical infrastructure components might compromise a testing session (besides others).;
- Bad or no communication of test parameters and/or SUT configuration parameters between the participants and associated software;
- Previous problems in the testing are not reported sufficient;
- Availability of the testing team: If not enough auditors are available to verify the tests performed, SUTs that are fully passing the tests from a technical perspective may be rated as successful too late, or worst, not rated at all;
- Subjective, different judgements by different person from the testing team: Since the professional background and the experience of person are (of course) different, they may judge differently whether a test has been passed or whether it failed. For the most common cases, the test result interpretation of auditors should be narrowed done as far as possible within the profile specifications and the documentation that is available for auditors as well as for the participants. However, a gray area of finally rating a test will always exist but its

effect can be minimized if there is a “standard protocol” to be followed if an auditor is unsure of a participant feels judged wrongly. This could be for example an authoritative commission at the testing event that decides in case of dubious judgements.

- Insufficient planning of logistic or resources. If for example the test management is not operative for the full time of the test session, no tests can be performed or documented in the worst case. Also, physical security of equipment is an actual issue for large-scale face-to-face testing events like the IHE Connectathons. Usually more than a hundred systems with powerful and expensive computing equipment are all day and partly all night stored in a large room or exhibition hall. Thus, there must be access controls and night shifts of security staff in order to guarantee security. Also, there should be guidelines available that cope with the competitive background of many systems in large test events. Often many vendors are involved and there may be incidents with parties trying to gain knowledge about other participant’s systems.

3.7.5 Test plan execution roles and responsibilities

The Test Manager is responsible for the test plan execution. He manages the testing team, the SUT operators and reports to the top level management.

3.8 PROCESS: Test Management

3.8.1 Why do we need test management?

The objective of this process is to define how the test plan definition and execution need to be managed. A sound management will lead to success by checking and controlling the progress of the testing process as well as reviewing the risk planning with the QA manager.

The testing interoperability process is a project on itself and thus should be basically led and managed as a classical project.

Test management commitment is essential for the implementation and ongoing success of the interoperability Quality Management System.

3.8.2 Objective for test management

The objective of the test management is to lead the testing process according to the test plan definitions and by respecting schedule, resources and budget.

3.8.3 Work to be done

The test plan definition defines the scope, features, tasks, criteria, resources and the test environment needed for a specific test sessions.

The Test Manager has the responsibility to define:

- The planning and roadmap for all the tasks of the process, eq. by using for example Gantt charts;

- The budget for performing the tests;
- The organisation of the project (roles and responsibility of each member of the testing team);
- The risk planning managed by the QA manager;
- Accounting and management report to follow the progress of the project;
- Reporting to the top level management.

The Test Manager has also the responsibility to:

- Communicate the test plan to the testing team;
- Manage and motivate the testing team;
- Check, monitor including definition of the quality indicators and review the progress of the project;
- Review and write the final report including the performance and efficiency of the project.

→ Define the responsibilities and the work to be done for the test manager in your organisation and related to a specific test case.

3.8.4 **Project management risks and failures**

The causes of the failures are classical as from common project management:

- Profile specification not stable or insufficient;
- Unrealistic planning;
- Testing team with no skills;
- Costs exceeding the budget;
- Risks underestimated for the project;
- Unclear roles and responsibilities.

3.8.5 **Project Management roles and responsibilities**

The Test Manager has the responsibility to manage the project with the support of the QA Manager. When the project is too small, the Test Manager is directly responsible of the quality of the project (he is also QA Manager).

3.9 PROCESS: Test Management Update

Test Management Update is the process of controlling the deployment and maintenance of all the testing process.

One activity has been identified:

- Audit and review

3.9.1 Why do we need a management update?

Because the test environment is not stable and is responding to the evolution of the test needs, the test management must be updated frequently. The update management is impacted by:

- Evolution of test plan definitions;
- Consequences of different kinds of unexpected events (e.g. lack of resources, technical problems, update of the interoperability specifications…) during the test planning such as any other projects.

The test update management depends on the testing cycle in the organisation (e.g. how many events are organised per year), how many test plan definition are defined).

The Test Management Update is part of the quality of the test environment. To improve the test plan definition, the organisation will analyse the quality indicators in order to evaluate and to update the testing process.

3.9.2 Objective for management update

Two objectives are to be met:

- Quality improvement of the testing process;
- To maintain stable and efficient test environment by updating the test plan definition in order to follow the life cycle of the interoperability specifications.

3.9.3 Work to be done

To improve the quality of the environment, an audit process has to be planned on a regular basis. The quality indicators and criteria are defined at the beginning of the testing process. The categories of criteria could be (example):

- Coverage of the tests
- Risks generated
- Faults detected

The indicators are analysed and an action plan is delivered.

To maintain the stability of the environment, for each activity a regular report is delivered by the test manager in terms of the evolution of the tasks, workload, resources and problems.

A review can be planned in order to verify that the testing process and activities are correctly implemented and operational. Criteria will be defined in order to prepare the review. One element supporting that is the regular report. The review should give a traceability of all actions that were planned since the previous review. This review is mostly under the responsibility of the QA Manager. Other responsible persons could be defined depending of the enterprise organisation.

→ Define the quality indicators and record it

→ Define a template of the regular report

→ Define formal process of review

3.9.4 Management update risks and failures

The first risk is to launch a test plan without taking quality indicators into account from the earlier stage. The second risk is not to define a clear and transparent review process. The general cause of failure is a bad communication to auditors in the review process.

In order to maintain a stable environment, a clear definition of the testing process and a well-educated testing team are crucial. When the environment has to change, it is important to check that the testing team understands this evolution very well and that the proposed changes can be integrated into the existing environment. That may include for example planning of additional training courses and decent communication.

3.9.5 Management update roles and responsibilities

The management update is under the responsibility of the Test Manager supported by the QA Manager. When the project is too small, the Test Manager is directly responsible of the quality of the project (he is also QA Manager).

3.9.6 Management commitment

The Top Level Management of the organisation is committed to continually improve the quality management system for interoperability. This is demonstrated by:

- communicating to the organisation the importance of meeting both, the customer’s as well as statutory and regulatory requirements;
- communicating the quality policy within the organisation;
- communicating the quality objectives within the organisation conducting management reviews;
- ensuring the availability of resources.

3.9.7 Customer focus

Top Level Management ensures that customer requirements for interoperability are determined and met with the aim of enhancing customer satisfaction.

3.9.8 Quality Policy

Top Level Management has to establish a quality policy that states the organisation’s position with regard to product quality.

The quality policy has to be appropriate with regard to the aims of the organisation and includes a commitment to comply with requirements. It continually improves the effectiveness of the quality management system.

The quality policy also provides a framework for establishing and reviewing quality objectives.

Top Level Management ensures that the quality policy is communicated and understood within the organisation, and is reviewed for continuing suitability.

3.9.9 **Planning**

Quality objectives:

- Quality objectives, including those to meet requirements for interoperability, are established at relevant functions and levels within the organisation.
- The quality objectives are measurable and consistent with the quality policy.
- Quality objectives are used as the primary tool of controlling a given process in terms of its performance and effectiveness.

Quality management system planning:

- Top Level Management ensures that the planning of the quality management system is carried out in order to meet all requirements as well as the quality objectives and in order to maintain the integrity of the quality management system when changes to the quality management system are planned and implemented.

3.9.10 **Responsibility, authority and communication**

Responsibility and authority:

- Top Level Management ensures that responsibilities and authorities are defined and communicated within the organisation.

3.9.11 **Management representative**

Top Level Management has to appoint a member of the organisation's management who – irrespective of other responsibilities – is responsible and has the authority in:

- ensuring that processes needed for the quality management system are established, implemented and maintained;
- reporting to Top Level Management about the performance of the quality management system and any need for improvement.

3.9.12 **Internal communication**

- Top Level Management ensures that appropriate communication processes are established within the organisation.
- Meeting minutes are maintained in order to ensure the effectiveness of the communications.

→ Define the internal communication for the testing process and associated QMS