Smart Open Services for European Patients

Open eHealth initiative for a European large scale pilot of patient summary and electronic prescription

D3.C.1 Appendix A - Revised Requirements of epSOS Testing Environment

WORK PACKAGE	**WP3.C**
DOCUMENT NAME	**D3.C.1 Appendix A - Revised Requirements of epSOS Testing Environment**
SHORT NAME	**D3.C.1 Appendix A**
DOCUMENT VERSION	**1.0**
DATE	**December 21, 2012**

Document Information

Project name	Smart Open Services – Open eHealth initiative for a European large scale pilot of patient summary and electronic prescription
Author/ person responsible	Mustafa Yuksel (SRDC)
Document name	D3.C.1 Appendix A - Revised Requirements of epSOS Testing Environment
Status	in process | submitted to QA | accepted by QM | **approved**

History of Alteration

Version	Date	Type of editing	Editorial
0.1	2012-01-31	Preparation of the first draft of epSOS Revised Testing Requirements	Mustafa Yuksel (SRDC)
0.2	2012-02-01	Updates by Marcello Melgara	Marcello Melgara (LISPA)
0.3	2012-02-10; 2012-02-14; 2012-02-14; 2012-02-20; 2012-02-26; 2012-02-28	Updates/comments by reviewers have been merged into a single document	Sören Bittins (FhG); Marcello Melgara (LISPA); Fredrik Linden (SALAR); Stefan Gustafsson (CEHIS); Monika Zemenova (IZIP); Mustafa Yuksel (SRDC)
0.4	2012-03-01	Update of the document according to all the comments received on top of version 0.2	Mustafa Yuksel (SRDC)
0.5	2012-03-06	Upon the suggestion by Mustafa, it has been agreed by the TPML to make this document official by making it an appendix of D3.C.1 - Proof of Concept Testing Strategy.	Mustafa Yuksel (SRDC)
0.6	2012-12-10	Many comments and feedback have been received from NEPCs, NEPC Tech Leads and also IHE-Europe during spring 2012. After completing the main document D3.C.1, these input are integrated into the document. The document title is changed from "epSOS Revised Testing Requirements" to "Revised Requirements of epSOS Testing Environment".	Artur Novek (EESTI); Gergely Heja (ESKI); Julia Silchenko (EESTI); Karima Bourquard (IHE); Mie H. Matthiesen (NSI); Monika Zemenova (IZIP); Mustafa Yuksel (SRDC); Norbert Repas (ELGA); Oliver Kuttin (ELGA); Steen Manniche (NSI); Stefan Gustafsson (SALAR); Stéphane Spahni (HUG)
1.0	2012-12-21	Updated according to the final quality review.	Jeremy Thorp (NHS); Mustafa Yuksel (SRDC)

Referring Documents

Date	Type	Description	Version	Origin	Document
2011-11-09	pdf	epSOS Description of Work	v.0.2	epSOS Consortium	DoW_epSOS5_PartB_20111109 .pdf
2010-10-15	doc	D3.9.2 - Testing Methodology, Test Plan and Tools	v1.0	epSOS WP3.9	D3 9 2_epSOS_Testing_Methodology _Plans_Tools_v1.0.docx
2011-07-17	doc	epSOS end-to-end Functional Testing for Projectathon and Pre Pilot Testing: Guidelines for HPs and PNs	V0.9	epSOS WP3.10	D3.10.1 “Result of Testing”, Appendix A8
2012-12-21	doc	D3.C.1 - Proof of Concept Testing Strategy	v1.5	epSOS WP3.C	D3.C.1 - Proof of Concept Testing Strategy v1.5.docx
2012-12-28	doc	D3.C.1 Appendix-B - Proof of Concept Testing Strategy Details	v1.6	epSOS WP3.C	D3.C.1 Appendix-B - Proof of Concept Testing Strategy Details v1.6.docx
2012-10-31	doc	Change management of epSOS Operation	v0.8	epSOS Change Manager (ChM), Change Advisory Board (CAB)	epSOS Change Management Master document.doc
2010-02	pdf	Feasibility Study for a Global eBusiness Interoperability Test Bed (GITB)	v1.0	The Global eBusiness Interoperability Test Bed initiative (GITB)	CWA16093TestBed-Phase1-Deliverable.pdf
2012-02	pdf	GITB Phase 2 Report - Conceptualization of the GITB framework and architecture	v1.0	The Global eBusiness Interoperability Test Bed initiative (GITB)	CWA_16408.pdf

Table of contents

1. Introduction

During the first phase of the epSOS Project, testing strategies, processes and tools were defined in D3.9.2, taking into consideration the complexity of cross-border interoperability in eHealth.

The main joint decision was to leave to PNs the role of performing Unit test and System Integration test, while to assign to an internationally recognised, independent entity, namely IHE Europe, the role of defining the test tool (based on Gazelle) and managing the test procedures and test event (the Projectathon, strictly connected to the Connectathon), which is also open to entities out of epSOS Consortium (i.e. vendors and institutions).

WP3.10 implemented the test procedures and provided new tools and processes to cover the end-2-end functional testing (D3.10.1 Appendix A8).

The aim of this document, built upon previous strategies, procedures, tools and gained practical experience, is to revise the requirements of epSOS testing environment. The document mostly concentrates on the necessary technical functionalities to be provided by the test bed (i.e. test execution environment, see the definitions in Section 1.1) that will be used in epSOS; however it also focuses on other aspects such as sustainability and maintainability.

The requirements that are presented here are created considering the specific cases of epSOS. On the other hand, these requirements, especially the technical ones, are also in line with the available results of the Global eBusiness Interoperability Test Bed initiative (GITB)[1].

GITB develops methodologies and architectures that support e-business standards assessment and testing activities, from early stages of eBusiness standards development to implementation and deployment of large-scale eBusiness solutions. GITB is hosted by the European Committee for Standardization (CEN) and supported by European, North American and Asian partners, such as NIST, KorBIT, OASIS, OAGI and industry associations, such as AIAG (Automotive Industry Action Group).

GITB's long-term objective is to establish a shared and global testing platform which allows for plug-in testing capabilities (such as test suites, specialized validation components, message adapters, etc.). These testing capabilities can be supported by existing Test Beds, by remote services or by future test components to be developed. In order to capture e-business testing requirements and validate the approach, GITB closely works with user communities in eHealth (HL7), automotive (long-distance supply chains) and public procurement (PEPPOL).

In Section 3, the document also presents some issues reported by PNs regarding the test cases and tools that are used in epSOS phase I. Finally, in Section 4, a summary of epSOS phase I experience regarding testing is presented.

This document does not aim to define epSOS testing phases (e.g. projectathon, pre-pilot testing), the interaction among these phases or the necessary conformance gates to participate to these phases. These information and further are already defined in D3.C.1 and D3.C.1 Appendix-B.

[1] The Global eBusiness Interoperability Test Bed initiative (GITB), http://www.ebusiness-testbed.eu

1.1 Testing Definitions

These definitions are adapted from the GITB project for epSOS purposes. For a more comprehensive list of definitions, please refer to the GITB reports that are referenced in the 3rd page of this document. GITB has already benefited from the existing definitions of standards by ISO, ETSI and ITU in order to provide such a comprehensive list of testing terms.

Conformance Testing: Process of verifying that an implementation of a specification (System Under Test [SUT]) fulfils the requirements of this specification, or of a subset of these in case of a particular conformance profile or level. Conformance Testing is usually realized by a Test Bed connected to the SUT. The Test Bed simulates eBusiness protocol processes and artefacts against the SUT, and is generally driven by the means of test scripts.

eBusiness Specification: An eBusiness Specification is any agreement or mode of operation that needs to be in place between two or more partners in order to conduct eBusiness transactions. An eBusiness Specification is associated with one or more of three different layers in the eBusiness interoperability stack: transport and communication (Messaging) layer, Business Document layer and Business Process (i.e. workflow) layer. In many situations, an eBusiness Specification comprises a set of standards or a profile of these. Within the scope of epSOS, eBusiness Specifications are the architectural documents that explain how existing profiles and standards are used for epSOS transactions (e.g. D3.4.2, D3.A.4), and also the documents that explain in detail the templates that should be used in epSOS electronic documents (e.g. D3.9.1 Appendix B1, B2).

Interoperability Testing: A process for verifying that several SUTs can interoperate at one or more layers of the eBusiness interoperability stack (see “eBusiness Specification”), while conforming to one or more eBusiness Specifications. This type of testing is executed by operating SUTs and capturing their exchanges. The logistics of Interoperability Testing is usually more costly (time, coordination, set-up, human efforts) than Conformance Testing. Conformance Testing does not guarantee interoperability, and Interoperability Testing is no substitute for a conformance Test Suite. Experience shows that Interoperability Testing is more successful and less costly when conformance of implementations has been tested first.

System Under Test (SUT): An implementation of one or more eBusiness Specifications, which are part of an eBusiness system that is to be evaluated by testing. Within the scope of epSOS testing, a SUT is always a National Contact Point (NCP), or a service provided by the NCP. In terms of European Interoperability Framework (EIF), epSOS NCP concept can be termed as a Pan-European eGovernment Service (PEGS) as well.

Test Bed: An actual test execution environment for Test Suites or Test Services.

Test Bed Provider: A general role that applies to anyone offering Test Bed Platform to Test Participants / Managers / Designers

Test Case: A Test Case is an executable unit of verification and/or of interaction with a SUT, corresponding to a particular testing requirement, as identified in an eBusiness Specification.

Test Report: Result of verifying the behaviour or output of one or more SUT(s), or verifying Test Items such as Business Documents. It is making a conformance or interoperability assessment (see Conformance Testing and Interoperability Testing). It is generally intended for human readers (although possibly after some rendering, e.g. HTML rendering in a browser or after a translation from XML to HTML).

Test Service: Service interfaces (e.g. as a Web service) that can be remotely accessed either by authorized users (e.g. via a Web browser, or via an editor) or accessed by another Test Bed. These Services are called “General” Test Services, as their interface provides for a coherent set of operations in a particular functional domain (e.g. test suite design, or test execution control).

Test Suite: A Test Suite defines a workflow of Test Case executions and/or Document Validator executions, with the intent of verifying one or more SUTs against one or more eBusiness Specifications, either for conformance or interoperability. It is a complete set of artefacts that is executable by a test bed/engine.

1.2 Requirement Definitions

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Requirements of epSOS Testing Environment

This section presents the identified requirements for epSOS testing environment. The requirements are organized in three main sections:

1. Basic Requirements
2. Engineering Level Requirements
3. Sustainability Requirements

2.1 Basic Requirements

This section focuses on common epSOS testing environment requirements.

1. ***Test bed actuality and synchronization with epSOS specifications:*** The test bed and the test suites MUST always be kept updated according to the latest approved epSOS specifications, according to the time plan agreed between WP3.C and Test Bed Provider, taking into account the epSOS Change Management Process. With approval, Project Steering Board (PSB) approval is RECOMMENDED; but in the worst case Technical Project Management (TPM) approval MUST be ensured. When there are updates in existing specifications, in order to ease the adaptation of the test bed and the test suites, the changes MUST be clearly identified. WP3.C Validation in cooperation with WP3.A Architecture, WP3.B Implementation and TPM MUST provide the correct versions of the specifications to the test bed provider.
2. ***epSOS testing scope:*** epSOS testing MUST be focused on cross-border data exchange, hence the NCP to NCP communication. In addition, epSOS testing includes end-2-end functional testing, semantic and document validation, and existence and formal compliance testing of audit trails / security provision as well. On the other hand, although NCP-to-NCP testing (including end-2-end testing) might reveal issues related with NCP to National Infrastructure communication on one side, or on both sides, this is not the main objective of epSOS testing.
3. ***Monitors' knowledge of specifications:*** The monitors (i.e. human testing experts) that are provided by the test bed provider MUST have, as a whole, thorough knowledge of the epSOS specifications well before any physical or online testing activity.
4. ***Test efficiency:*** Although fine-grained test suites for testing very simple transactions SHOULD be in place, in general, the test bed and the test suites MUST always test as much as possible in a single step, to reduce efforts on the testers' and also on the validators' (i.e. monitors) side.

    4.1. ***Test efficiency example:*** For example, while testing a patient summary retrieval transaction implemented via IHE XCA, the test bed MUST perform validation checks at the messaging layer (SOAP header, security assertions in the SOAP header, SOAP body, SOAP attachments), at the document layer (structural validation of the PS document, semantic validation of the PS document) and also related with secure communication (i.e. use of certificates over HTTPS and establishment of the VPN communication) at once. The testers MUST NOT be forced to do several tests via several different test components/interfaces as in the case of epSOS Phase 1 testing process, while all can be done in a single test execution.
5. ***Interoperability testing:*** epSOS testing MUST NOT be limited to only conformance testing against some simulators; it MUST also include interoperability testing. When using a non-secure communication mode (no VPN and TLS) or a limited secure mode (TLS with private keys shared with the test bed in advance), the test bed MUST have the capability to listen

the communication among the SUTs and automatically perform validation checks on the intercepted messages. For details, please see sub-section on "Interoperability Testing Requirements" under Section 2.2.

6. ***Assurance of patient safety:*** epSOS testing MUST assure patient safety. Hence, focus MUST be set on semantic transformation, CDA document scrutiny and end-2-end functional testing as being of extreme relevance to assess the robustness with respect to patient safety protection.
7. ***Assurance of security & privacy:*** epSOS testing MUST assure information security, privacy rights and full compliance to the data protection regulations. Hence, security robustness is an attribute of the SUT that MUST be covered by testing procedures, including the correctness of non-repudiation functionalities.

2.2 Engineering Level Requirements

This section focuses on engineering (i.e. technical) level requirements of the test bed to be used in epSOS. The main categories in this section are:

- Document Layer Validation Requirements
- Messaging Layer Validation Requirements
- Business Process (Workflow) Layer Validation Requirements
- Interoperability Testing Requirements
- Common Technical Requirements of the Test Bed

Document Layer Validation Requirements

1. ***Syntax & structure validation:*** The test bed MUST perform document syntax and structure validation of the epSOS electronic documents, i.e. electronic Patient Summary (ePS), ePrescription, eDispensation, eConsent, Audit message (which can be considered as a document in epSOS context) and any further documents to be designed during the second phase of epSOS, according to the related structured document schemas, such as the XSD of HL7 CDA R2.
2. ***Semantic validation:*** The test bed MUST perform semantic validation of the epSOS electronic documents according to the content templates and business rules that are defined within the epSOS specifications on top of the common schemas of the used standards. The restrictions that are defined verbally in these content templates and business rules MUST be expressed in a machine executable way by the test bed provider and validated by epSOS experts responsible for related epSOS specifications. Since all epSOS documents are XML-based, the RECOMMENDED way of expressing these business rules is using Schematron or model-based validators.
    2.1. ***Mandatory/optional attribute validation:*** The test bed MUST validate mandatory/optional attribute constraints, which are defined by epSOS content templates and business rules in addition to similar constraints in the XSDs.
    2.2. ***Coded concept validation:*** The test bed MUST check whether the coded data fields comply with the defined epSOS value sets and additional vocabularies. Validation MUST cover code and code system identifier attributes; in case the attributes for code system name and designation are included in validation, they MUST be case insensitive.

Messaging Layer Validation Requirements

1. ***Conformance of communication:*** The test bed MUST perform validation of the exchanged messages according to the used messaging protocol. In epSOS, currently all communication among NCPs is done via SOAP Web Services. The test bed MUST be able to check conformance of NCP communication according to the Web Service specifications as restricted by the preferred interoperability profiles such as IHE XCPD, XCA, XDR and XUA. These profiles define both structural and semantic requirements, all of which MUST be covered.
    1.1. ***SOAP header validation:*** The test bed MUST check whether the SOAP header is valid. This also involves checking the syntactic and semantic validity of the security/privacy assertions (TRC assertion, HP assertion) that are exchanged within the WS-Security headers.
    1.2. ***SOAP body validation:*** The test bed MUST check whether the SOAP body is valid. In some transactions such as XCPD or XCA:list responses, all the requested information is presented directly within the SOAP body. The structural and semantic validity of the SOAP body MUST be checked.
    1.3. ***SOAP attachment validation:*** If utilized by an interoperability profile, the test bed MUST check whether the SOAP attachment(s) are valid. In some transactions such as XCA:retrieve, the requested patient document is transported as a SOAP attachment. As explained in the 4th basic requirement (i.e. test as much as possible at once), after getting the attachment, the messaging layer MUST call the document layer validation facilities for a complete structural and semantic validation of the document.
2. ***Validation of secure communication:*** The test bed MUST check compliance to secure communication requirements as defined by the epSOS specifications, i.e. use of security certificates over HTTPS and establishment of the IPsec VPN connection.

Business Process (Workflow) Layer Validation Requirements

1. ***Automated workflow testing:*** The test bed MUST be able to perform automated workflow (i.e. business process) tests. An example workflow constitutes of the following steps: identification of the patient, defining remote consent, patient summary document query and patient summary document retrieval. Such workflows will be defined by epSOS.
    1.1. ***Workflow execution:*** The test bed MUST ensure the sequence of steps that need to be realized for a workflow, and wait for relevant requests/responses from the SUT, or send relevant requests/responses to the SUT. The status of the workflow execution MUST be presented to the tester, who can be a technical person or health professional according to the workflow definition, via graphical user interface (GUI).
    1.2. ***Reuse of existing test resources:*** At the workflow layer, the test bed MUST benefit from the already existing validation facilities at the messaging and document layers. Again, this MUST be in line with the 4th basic requirement (i.e. test efficiency).

Interoperability Testing Requirements

This section presents some details about the 5th basic requirement (i.e. interoperability testing) defined in Section 2.1.

As already explained in the definitions section, interoperability testing is a process for verifying that several SUTs can interoperate at one or more layers of the eBusiness interoperability stack (see “eBusiness Specification”), while conforming to one or more eBusiness Specifications. This type of testing is executed by operating SUTs and capturing their exchanges automatically.

According to GITB, the three most common ways to capture message traffic between SUTs are:

a) Using a “man-in-the-middle” system operating and re-routing messages at transport level (e.g. an HTTP proxy or a TCP intermediary). This is typically the least intrusive approach, although it imposes restrictive conditions (the messages and sessions MUST NOT be encrypted; or if they are encrypted, the test bed as the intermediary MUST have the necessary private keys to decrypt them).

b) Instrumenting of one of the SUT so that message capture is performed at the endpoint, e.g. on the message handler of the SUT. Later on this message capture can be consolidated in a Test Execution Log.

c) Configuring the sending SUT(s) so that they duplicate messages sent and forward a copy a Monitoring component or directly to the Test Bed.

Ideally, the message capture SHOULD NOT interfere with the way the SUTs interoperate as they would under real business conditions. Therefore, the best alternative in the above list is a).

Figure 1 presents briefly how automatic capturing of message exchange between the SUTs, and then automatic validation of the captured messages, documents and interactions can be done.

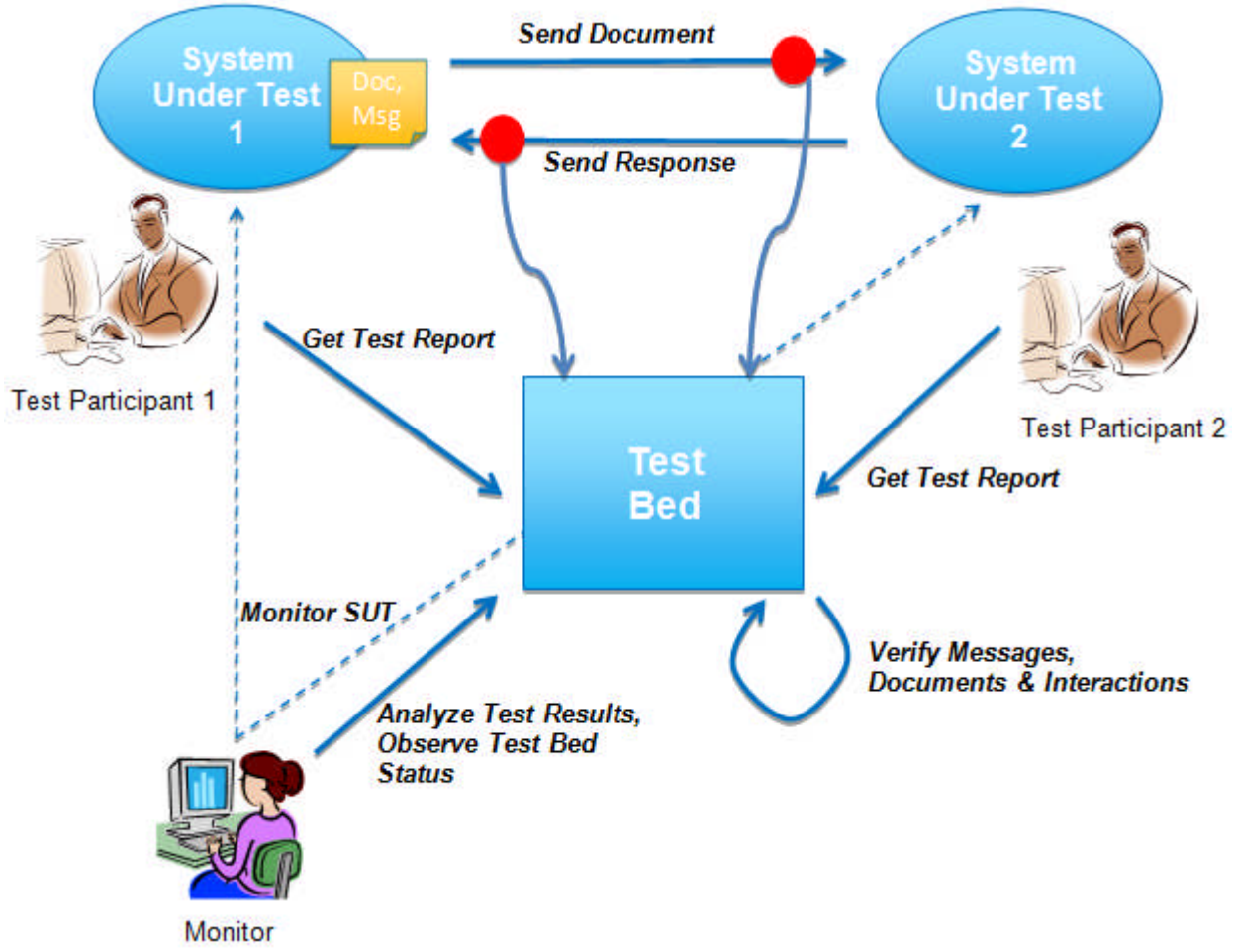

Figure 1 Basic Interoperability Testing [Adapted from the GITB Project]

The test bed MUST benefit from the already existing validation facilities at the messaging and document layer for realizing interoperability testing. Therefore, the main issue to be concentrated here is the automatic capturing of the communication between SUTs, by at most requesting the SUTs to change their endpoints for sending the messages; i.e. a SUT can send a message to the test bed which then forwards this message to the intended SUT, instead of directly sending the message to the intended SUT. The test bed and the interoperability test suites MUST provide the testers necessary configuration options via a GUI.

Common Technical Requirements of the Test Bed

1. ***Registration of testers:*** The test bed MUST provide the capability to the testers to register themselves and their SUTs for testing activities. There MUST be testing sessions dedicated to epSOS testing activities.
    1.1. ***Alternative user authentication means:*** The test bed MAY allow Open ID and STORK eID integration for testing accounts.
2. ***Accessibility of test suites:*** The test suites MUST be clearly identified and easy to locate and initiate. In case the location of a test suite changes, then all previous links pointing to it MUST be updated and informed to the testers immediately.
3. ***Interaction with tester:*** The test bed MUST have the ability to inform the tester about the testing steps to be executed within a test suite, before initiating that test suite. Testers MUST be offered a test suite description that can be narrative text or some graphical representation describing the test suite flow and each step, also explaining any required information from the tester or the SUT before starting the execution of the test. For complex workflows, it is RECOMMENDED to have a notification mechanism (over the user interface) that informs the tester about the status of execution.
4. ***Adaptive test execution parameters:*** The test bed MUST provide a mechanism to ask the testers some necessary parameters and information about their SUTs before starting test execution; related to network configuration, security configuration or some epSOS service specific configuration such as providing home community id of a PN.
    4.1. ***Binding automation:*** This mechanism MUST automatically bind the information into the test suite without any manual intervention so that the corresponding message/document elements can be tested if they are consistent with provided information.
    4.2. ***Minimalist approach:*** Such configuration parameters requested from the testers MUST be kept at minimum.
    4.3. ***Remembering configuration parameters:*** For ease of use, it is RECOMMENDED that the GUI of the test bed remembers previously provided configuration parameter values, at least the last ones, across testing sessions (session as in user session utilized in Web applications, not epSOS test sessions such as projectathon). In parallel, it is also RECOMMENDED that the testers enter their SUT configuration once for an entire testing session and the configurations are bound to the testing sessions.
5. ***Exchange of SUT configurations:*** Both interoperability and conformance test scenarios necessitate some configurations regarding SUTs. The test bed SHOULD enable testers to share configuration parameters of their SUTs among each other.
6. ***Reporting of results:*** The test bed MUST have a reporting mechanism that clearly informs the testers about the results of the testing executions they had.

6.1. ***Summary view:*** The test report MUST have a summary view presenting the overall success/failure status of the fine-grained test steps.

6.2. ***Accessibility of report details:*** The testers SHOULD be able to access parts of a test report easily, in which details about the results of a test step are presented.

6.3. ***Error location and evidence:*** The test report MUST clearly point to the location of any error of the SUT, supporting it with evidence about the error.

6.4. ***Error resolution guidance:*** In case of presence of an error detected in the SUT(s), the test report SHOULD also provide information, as much as possible, on how to resolve that error.

6.5. ***Error severity classification:*** Severity of the detected errors SHOULD be included in the test report. Defect severity classification is explained in Section 3.6.3.1 of D3.C.1 Appendix-B.

7. ***Utilization of a test definition language:*** In order to facilitate easy maintenance of the test cases and test suites, the test bed SHOULD adapt a structured test definition language for defining the test cases and test suites. The test bed SHOULD be able to automatically execute instances conforming to this test definition language.
8. ***History view:*** The test bed MUST provide a history view to the testers providing the complete history of test executions for their registered SUTs. This view MUST provide basic statistics as well. At any time, it MUST be possible to see the detailed test report of a specific test execution.
9. ***Management GUI:*** Similar to the history view provided to the testers, the test bed MUST also provide a management GUI to be used by monitors of the test bed provider and authorized persons in epSOS (e.g. coordinator, WP3.C leader, etc.; to be decided). The management GUI MUST provide the testing history of all the SUTs supported with basic statistics and the functionality to access the details of any test execution by any SUT. User friendly filtering mechanisms SHOULD be in place.
10. ***Versioning:*** Test bed, test suites, their significant parts such as schematrons and test data MUST be versioned. The version MUST be visible to the testers in the GUI and for each version there SHOULD be a statement to which version of epSOS specification it corresponds and a log of changes since the last version.
11. ***Validation of the test bed by experts:*** The test bed MUST be validated by experts responsible for the relevant specification before each testing session, in case a major revision in specification or test bed was done since the last testing session. The validation SHOULD be done, e.g., by testing with the reference implementation (in epSOS Phase 1), systems of PNs that have already achieved a PPT-slot or reference set of documents. The test report from this validation MUST be made available to the testers.
12. ***API availability:*** In addition to GUIs to be used by the non-technical end-users, the test bed SHOULD provide an API to interact with test suites, test progress and test reports for enabling totally automated test execution.
13. ***Linking with requirement/issue management tools:*** The test bed MAY have linking functionality to some issue management tools such as such as Redmine and JIRA, and requirement management tools such as Contour, which are used by epSOS as well.

2.3 Sustainability Requirements

This section focuses on requirements of epSOS testing environment from the sustainability point of view. Sustainability can be split as process sustainability and test bed sustainability.

Regarding the testing process sustainability, the following requirements apply:

1. ***Compliance validation by independent bodies:*** In line with the implementation strategy defined in D3.B.1 and the sustainability strategy under definition in WP2.2, which foresees an independent growth of vendors and PN level solutions, the procedure of validating the compliance of NCPs and components to the epSOS specification SHOULD become more and more a task assigned to an independent body that can organise and manage test events open to all, and provide validation of conformity.
2. ***Sustainability of test procedures and events:*** Test procedures and events SHOULD become self-sustainable with the attendees paying their registration fees fully or partially, in order to free them from the need of project funding.

As for test bed and test bed provider sustainability, the following requirements apply:

1. ***Compliance to an interoperability testing framework:*** epSOS testing strategy and the test bed to be used in epSOS SHOULD be compliant with an internationally recognized interoperability testing framework.
2. ***Proven experience of test bed provider:*** The test bed provider MUST be capable from the technical and global organizational point of view. They MUST have experience in this area, proven with previously organized similar physical and online interoperability testing events.
3. ***Long term availability of epSOS testing resources:*** epSOS test suites and the test bed SHOULD be available after the end of epSOS project, i.e. after 2013. It is also RECOMMENDED that such testing resources are available as part of the regular work of the test bed provider; not only as a derivative.
4. ***Documentation of testing facilities:*** The test bed provider MUST provide guideline documents and also videos for using the testing facilities provided by the test bed and test suites.

3. Reported Issues

This section presents the issues reported by some PNs regarding the testing process, the set of test tools and test cases used in epSOS phase I. The reported issues are organized into different categories as major, medium and minor issues. These feedback by the PNs who actually had experience with epSOS phase I testing processes are really important, and they influence the revised testing requirements that are presented in the previous section.

Please note that the names of the PNs who reported these issues are not disclosed in this document, but if needed, the WP3.C Leader keeps track of the original reports.

Major issues:

- Gazelle by IHE does not make testing easy enough. For example, after exchanging some documents or patient information via XCPD and XCA requests/responses, the testers on both side have to manually find, copy and paste hundreds of lines of transaction logs from the NCP logs (might easily reach to hundreds of MBs) to very simple Gazelle Web forms. Then, it is almost impossible for the monitors to fully validate these hundreds of lines of transaction logs; they cannot import all these logs into Gazelle validators. This is not interoperability testing.
    - Gazelle has some proxy support for intercepting the natural way of exchanging documents on-the-fly; so that the testers can get rid of this time-consuming copy/paste loop. The validation of the transactions would be automatic as well in this case, via calls to the already existing Gazelle EVS validators. The use of proxy mandates some limitations in secure-mode communication though, as explained in the previous sections.
- Mostly manual, non-interactive, and exhaustive tests based on intercepted data sets instead of comprehensive, integrative scenarios and test simulators. This is both resource and time consuming.
- Test material is lagging behind the most recent epSOS specification updates, for instance during Pisa PAT D3.4.2 v1.00 was considered valid by IHE-Europe, while the PN implementations were already based on D3.4.2 v2.00.
- A significant part of the required test data needs to be extracted from sensitive sources, like the audit repository with no “efficient” interface → PNs need to find someone to actually gather the data in a manual process, which is expensive and requires a lot of effort (in particular if 2 or more parties are involved).
- Tests are not self-contained, in many cases a “real” partner is required in order to pass → When something goes wrong, it is not always clear whose fault it was.
- Gazelle tests do not check the whole epSOS medical information exchange workflow. There is no link between fine-grained steps of testing, e.g. from XCPD to XCA.
- Gazelle tests do not cover the security aspects of the transactions, i.e. HTTPS, VPN. For example, only with Bern PAT in May 2012, the XCPD and XCA initiating simulators started to support sending queries to HTTPS endpoints.
- Although the tests are designed and partially performed “on-line”, the validation is a completely manual process with significant delay between performing the test and their particular validation. IHE was verifying tests weeks later most of the time.
- Since the validation of the completed tests is very time and resource consuming on the IHE side, the number of verified tests is always quite below the total number of completed tests

by PNs; less than 70% of the tests were verified at Bratislava and Pisa Projectathons, although it took several weeks for IHE to send the testing reports.

- Gazelle tests as made available to epSOS participants are not suitable for continuous and automatic implementation tests → Total detachment from development and no immediate result for supporting the implementation activities of epSOS. [IHE-Europe notes that the validation services are actually implemented as Web Services, and they can be used for automated testing]
- epSOS PPT process is not fully sustainable → Every change requires a new roll-out of the tests, a full re-test of the whole solution, and constant monitoring (paid) is required to perform such tests with a huge effort.
- When we look at the actual piloting status, it can be concluded that the testing process and Gazelle tests were not very successful in evaluating the quality of the NCP implementations of the PNs.
- During epSOS phase I testing sessions, almost all the PNs have observed that the testing environment, bundled with epSOS specific test cases, and provided by IHE had some problems. Software applications usually have problems but the issue here is that IHE is generally late in providing maintenance support to resolve the issues reported by the users.
- The testing costs are too high both for epSOS and for PNs.
- Not directly related with IHE or Gazelle, but related with the overall testing process and the implementation strategy: Sometimes, the problems that are promised to be fixed in the upcoming version of a Common Component are not fixed actually; the problems still persist. There should be a way to ensure the quality of the Common Components before providing them to the PNs for installation and tests; and this can be done by proper testing of Common Components before releasing.

Medium issues:

- Some IHE monitors are not “epSOS-aware”, but expect IHE baseline transactions and flavours → epSOS-specific parts unknown.
    - For example, in Pisa, the PNs who were relatively fast in completing tests had to first explain to IHE monitors what a specific test is about, what needs to be done for passing this test; and then the monitors tried to validate these tests.
- IHE Gazelle tests are originally designed to be “interoperability” tests, however, in epSOS they have been repurposed for posing as conformance / security test → inconclusive.
- Schematrons are able to not only check the syntax and the “well-formed” semantics but also to check the plausibility and correctness of the contained information → include in-depth content conformance checks.
- Some tests are designed in a “failure” fashion, requiring input data that is faulty on purpose → epSOS systems cannot generate that data so tests are done “by hand” with very limited value but excessive effort.
    - Such failure tests can be achieved for document-level testing. But, one cannot simply order an HSM, SSCD or any other strong security component to issue "wrong" assertions/certificates/signatures etc. These require a massive amount of work, with very minimal achievements in the end.

Minor issues:

- Gazelle was not always the most stable tool to perform tests with → crashes, lost test data in one instance, overwriting of former results.
- Tests have not been tailored for the specific PN, or more importantly for specific transactions that the PNs support, so even for PNs with no eP/eD the respective tests have been displayed and were shown as incomplete.

4. WP3.10 Summary of epSOS Phase I Experience

- IHE Gazelle demonstrated rigidity and cumbersome usability that should be improved in order to make the use easier and more user friendly. Complaints from PNs are justified by these limitations.
- Bratislava and Pisa PAT have represented dramatic improvements in the NCP interoperability: progresses observed in a few days were hardly achieved with months of previous activities.
- Bratislava PAT has allowed to reach a first level of interoperability, consolidated and validated at Pisa PAT.
- Pisa PAT has allowed to validate interoperability and highlight the semantic interoperability issues.
- A bottleneck encountered at PATs and Connectathon were the limited performances of the network and internet infrastructure. It MUST be improved in the next PAT and Connectathon, in particular if remote-PAT will be introduced.
- PAT monitors MUST be more deeply trained to epSOS specifications.
- The basic problem encountered in PPT was that a test slot was not used for validating the end-2-end system, but to debug the Common Components and their integration with the National Infrastructure: this was not foreseen by the epSOS phase 1 Testing Strategy, according to which the unit tests and the integration tests MUST be done by PNs and vendors before the PPT; pre-PPT MUST be the proof of readiness.
- Debugging tools SHOULD be made available to PNs and vendors to effectively perform the unit and integration tests, while PAT and PPT MUST be devoted to validation by an independent and internationally recognised body. PAT and PPT results SHOULD be considered as third party conformance check, to be used both as procurement acceptance procedure and as epSOS conformance gates.