Integrative Cancer Research
INTEGRATE
Through Innovative Biomedical Infrastructures

SEVENTH FRAMEWORK PROGRAMME

ICT-2010-270253

INTEGRATE

Driving excellence in Integrative Cancer Research through Innovative Biomedical Infrastructures

STREP
Contract Nr: 270253

Deliverable: 6.2 Evaluation and validation procedures for the INTEGRATE environment

Due date of deliverable: (30-04-2012)
Actual submission date: (MM-DD-YYYY)

Start date of Project: 01 February 2011 Duration: 36 months

Responsible WP: UPM

Revision: final

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
David Perez-Rey	UPM	dperez@infomed.dia.fi.upm.es
Alejandro García-Ruiz	UPM	agarcia@infomed.dia.fi.upm.es
Jasper Van Leeuwen	Phillips	jasper.van.leeuwen@philips.com
Kristof De Schepper	Custodix	kristof.deschepper@custodix.com
George Manikis	FORTH	gmanikis@ics.forth.gr
Ioannis Karatzanis	FORTH	karatzan@ics.forth.gr
Víctor Maojo	UPM	vmaojo@fi.upm.es
Raúl Alonso	UPM	ralonso@infomed.dia.fi.upm.es

0.2 Documents history

Document version #	Date	Change
V0.1	01-02-2012	Starting version, template
V0.2	20-03-2012	Definition of ToC
V0.3	14-06-2012	First complete draft
V0.4	28-06-2012	Integrated version (send to WP members)
V0.5	28-06-2012	Updated version (send PCP)
V0.6	28-06-2012	Updated version (send to project internal reviewers)
Sign off	12-07-2012	Signed off version (for approval to PMT members)
V1.0	12-07-2012	Approved Version to be submitted to EU

0.3 Document data

Keywords	
Editor Address data	Name: David Pérez del Rey Partner: UPM Address: Facultad de Informática Universidad Politécnica de Madrid Campus de Montegancedo, s/n 28660 Boadilla del Monte, Madrid, Spain Phone: +34 91 336 74 45 Fax: E-mail: dperez@infomed.dia.fi.upm.es
Delivery date	

0.4 Distribution list

Date	Issue	E-mailer
28-06-2012	V0.6	fp7-integrate@listas.fi.upm.es

TABLE OF CONTENTS

1 INTRODUCTION

The main goal of this document is to identify the evaluation and validation procedures, required to test the INTEGRATE platform and environment. Such procedures are defined in the context of a quality assurance process, and based on a series of software quality principles. The first part of this document includes the definition of such quality principles, followed by metrics and criteria that must be applied within the evaluation and validation tests. In order to set the foundations of such process, the different environments that compose the platform are identified; specifying related verification and validation software services for the software requirements of the platform.

Once all the prerequisites of the procedures are defined, the main contribution of the document focus on the corresponding definition of the evaluation procedures within the INTEGRATE environment. Seven blocks have been defined: (1) security, (2) informed consent, (3) trial management, (4) screening scenario, (5) cohort selection, (6) central pathology review and (7) analytical tools. Blocks that are closely related to use cases defined at D1.4.

Each block corresponds to a series of use cases and should satisfy requirements and tests defined by the quality assurance procedures and general principles of software quality. It must be taken into account that such procedures depend on functionality related to each use case, thus, evaluation defined in this document are different for each block. Finally, the conclusion section summarizes the main guidelines of the evaluation procedures within the INTEGRATE environment.

2 QUALITY ASSURANCE PROCESS

2.1 GENERAL PRINCIPLES OF SOFTWARE QA

The concept of software quality concerns both companies and individuals dedicated to software development. Software needs to be certified and accredited during the development process in order to provide a high quality product. This document is based on the ISO 9126[1] quality model (concretely established at ISO 9126-1), widely adopted and identifying six main quality characteristics (Figure 1):

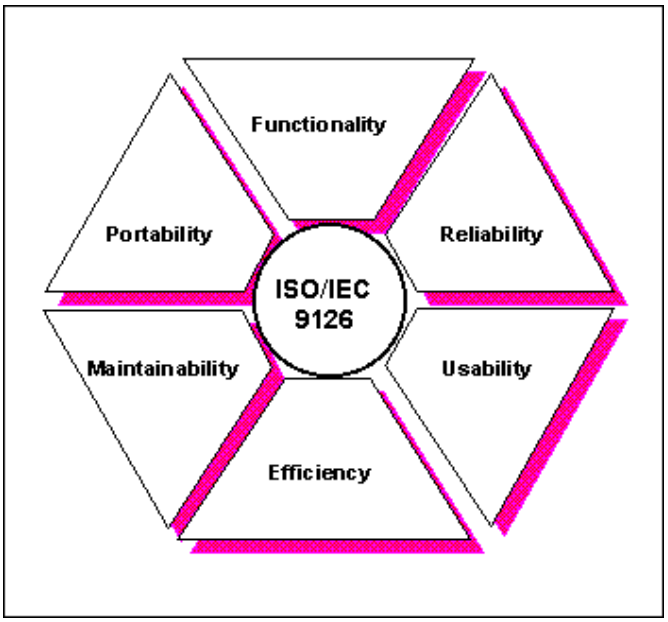

Figure 1 - The six quality characteristics of software

Five of these attributes (Reliability, Usability, Efficiency, Maintainability and Portability) will only exist if the *Functionality* (first quality characteristic) is provided. Bellow a brief description of the properties is included:

1.- Functionality: The degree in which software meets the requirements: *"are the required functions available in the software?".* Functionality is detailed by the following sub-attributes:

- **Suitability:** The appropriateness of the software functions for the specified tasks
- **Accurateness:** The proper working of the functions (giving correct results)
- **Interoperability:** The ability to interact with specified other components or systems

[1] http://www.cse.dcu.ie/essiscope/sm2/9126ref.html

- **Compliance:** Adherence to required standards/conventions/regulation of the developed software
- **Security:** The extent to which the software prevents unauthorized access to the software and the data.

2.- Reliability: The amount of time that the software is available for being use. Detailed by the following sub-attributes:

- **Maturity:** Frequency of failure
- **Fault tolerance:** Ability to recover from a component or an environment failure
- **Recoverability:** The capability of coping with and recovering from component/environment failures.

3.- Usability: How easy is it for users to interact with the system. Detailed by the following sub-attributes:

- **Understandability:** How easily can the provided functionality be understood by the users
- **Learnability:** The effort required for the users to learn to interact with the system
- **Operability:** Software's ability to easily operate in a given environment

4.- Efficiency: The degree to which the software makes optimal use of system resources. It is detailed by the following sub-attributes:

- **Time behavior:** Response rates and processing times of the functions
- **Resource behavior:** The amount of resources used and the duration of time that they spend using a resource while performing a function

5.- Maintainability: It refers to the simplicity to perform a modification. Maintainability is detailed by the following sub-attributes:

- **Analyzability:** Ability to identify the main cause of a failure in the software and to identify what component to change
- **Changeability:** The efforts needed to make a change in the software
- **Stability:** The risk of an unexpected effect derived from changes in the system
- **Testability:** Efforts needed to validate a system change

6.- Portability: The effort to move the software from one environment to another. Portability is detailed by the following sub-attributes:

- **Adaptability:** Ability of a system to get adapted for use in different environments
- **Installability:** The efforts needed to install software in a particular environments
- **Conformance:** It refers to the compliant capability of the software related to portability
- **Replaceability:** The ease to exchange a given software component

All these attributes may vary between different software products. Nonetheless, the standard provides an environment that allows organizations to define a quality model for a software product.

2.2 QA METRICS AND CRITERIA IN INTEGRATE

Different quantification metrics will be applied to assess the quality of the INTEGRATE platform components. Below, five perspectives are detailed on which we base the validation procedures of the INTEGRATE environment. From the effectiveness and performance perspective to the usability of the tools and services developed, the evaluation and validation procedures will describe planned tests and will be focused on blocks based on use cases.

The evaluation criteria will be periodically adapted to the current state of the development, considering end-user scenarios and clinical pilots as the general guideline of the validation/evaluation procedure. The validation of the platform will be conducted by the design and execution of test cases, with known results. Functionality, usability and maintainability, and also analysability and changeability will be the key criteria in the evaluation process. The outcome of these procedures will evaluate and monitor the adequacy of the INTEGRATE developed software with respect to its intended goal.

2.2.1 Development of ICT tools and services

In order to support activities for data sharing and collaborative environments, related ICT tools, infrastructures and services must be evaluated. User scenarios have been defined by clinical users to describe how data is shared using the required tools and services. The evaluation of such developments should include a comprehensive set of tests, evaluating the performance of the INTEGRATE tools and infrastructures. The main goal is to measure capabilities of the proposed solution within the clinical network of the project in terms of functionality, response time and ease of use.

A set of tests will be defined to evaluate the usability of the collaborative environment of the platform, i.e. to measure the ease of use. Expert surveys and comparisons with the available solutions will be carried out.

Finally, tools and services related to the predictive modelling framework, designed in the context of the WP5, will also require evaluation. To guarantee the potential in the

definition of predictive scenarios, therapy outcome comparison will be performed. It will be required to test if the proposed approach allows sharing predictive models in the area of breast cancer.

2.2.2 Effectiveness with existing infrastructures

Interoperability with existing clinical trial management systems and electronic health record systems of the clinical partners will be evaluated. INTEGRATE applications and services will be tested to store, manage, retrieve and analyze clinical trial management and the corresponding patient data. Such tests will include procedures such as a battery of queries and comparing results with existing systems.

2.2.3 Compatibility with existing methods and standards

To avoid building an ad-hoc solution, only fitting the involved clinical partners, a series of scenarios and tests to measure the compatibility and adherence of the platform with international standards will be defined. Such tests include a comprehensive analysis on biomedical terminologies that compose the core dataset of the platform (e.g. SNOMED-CT, LOINC, MedDRA or ICD or NCI Thesaurus, described on D2.1) and common data models (e.g. i2b2, OMOP or HL7-based models, described on D3.1). The final goal in this topic is to achieve compliance of the platform to such standards.

2.2.4 Adoption of the solution and popularity

Although the adoption of the proposed solution depends largely on external factors, the evaluation methods defined within this document will try to minimize technical limitations that may prevent adoption from external institutions. A loosely coupled and distributed architecture and the adoption of international standards will be evaluated in this regard. Impact of dissemination activities at main forums will also be essential to evaluate the potential adoption of the solution.

2.2.5 International cooperation

Collaboration with similar initiatives within the area is the main objective to evaluate international cooperation, including non-European biomedical user organizations. The objective is that our solution can interoperate with such initiatives, and again a modular architecture and standard adoption will be the main focus for such evaluation. Such cooperation includes contact with initiatives such as caBig, Transmart, i2b2 or OMOP (described at D2.1, D2.2 and D3.1).

2.3 INTEGRATE ENVIRONMENTS

In order to ensure a sufficiently stable INTEGRATE platform, we introduce over time three different environments (see Figure 2): the *development* environment, the *stage*

environment and the *pre-production*[2] environment. The environments are strictly disjoined, the components running in a particular environment are not allowed to access components/services running in a different environment (unless permission by the Lead Architect has explicitly been given).

The development environment is the environment in which – surprisingly enough – all development activities take place. Most, if not all, changes to the components will take place in the development environment. This implies that this environment might not be overly stable, as component interfaces (services) and their implementation may change over time.

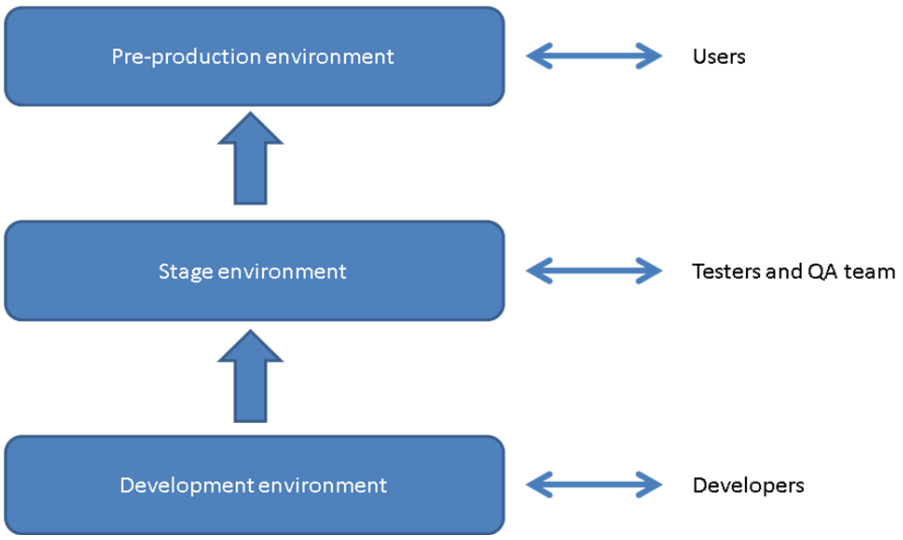

Figure 2 - INTEGRATE environments

Once a stable version of a component has been developed and tested, it can be deployed in the stage environment. The goal of the stage environment is to verify the proper functioning of the component in conjunction with the other components in the staging environment. In this environment, the QA team can perform technical verification and validation procedures to ensure that the components do the things right (verification) and that the components do the right things (validation).

When the proper functioning has been verified, the version of the component can be deployed in the pre-production environment, where it is available for users in the INTEGRATE consortium. In the preproduction, the users will use the platform to perform their work according the scenario's they have defined.

Of course, it is overkill to immediately have the three environments available. We will start with the development and stage environment, and intend to have the pre-production environment end of year 2 (as soon as necessary).

[2] The name "pre-production" suggests the existence of a fourth environment ("production environment"). This production environment is currently out of the scope of the INTEGRATE project, but is intended to facilitate exploitation of the platform.

2.3.1 Component migration procedure

In this section, the procedure is described for allowing the deployment of a version of a component in the different environments.
Table 1 specifies whose permission to obtain before deployment in a particular environment is allowed.
Table 2 shows who to inform before deployment in an environment.

The rules for deploying a component version differ for changes to the interface (provided by the component) versus changes to (only) the implementation.

	Component interface change	Component implementation change
Deployment in stage environment	• Developers of all components directly depending on this component (e.g. requiring the component interface) • Architecture team • QA team	• QA representative
Deployment in pre-production environment	• Developers of all components directly depending on this component (e.g. requiring the component interface) • Architecture team representative • QA representative	• Architecture team representative • QA representative

Table 1 - Required permissions

	Component interface change	Component implementation change
Deployment in development environment	• Developers of all components directly depending on this component (e.g. requiring the component interface). • Architecture team	
Deployment in stage environment	• Developers of all components directly depending on this component (e.g. requiring the component interface). • Architecture team representative • QA representative	• Developers of all components directly depending on this component (e.g. requiring the component interface). • QA representative
Deployment in pre-production environment	• Developers of all components directly depending on this component (e.g. requiring the component interface).	• Developers of all components directly depending on this component (e.g.

	• Architecture team representative • QA representative	requiring the component interface). • Architecture team representative • QA representative

Table 2 – Inform

It might be necessary (e.g. in case of malfunctioning of a component version) to roll back the deployment of a component version. In this case, the same procedures are required to roll back the version.

2.4 Software Repository requirements

The INTEGRATE consortium does not require partners to use any particular source versioning system, though Philips facilitates an SVN[3] repository which partners can use if they desire.

The requirement on the used source versioning system is straightforward: it should allow retrieving any component version every deployed in one of the environments.

2.5 Software verification and validation services

For notification and tracking of features and bugs, a (gforge[4]) tracker is provided by Philips.

Deliverable 6.3 will specify the evaluation and validation scenario's for the different components". When the evaluation and validation scenarios have been specified, it will be decided whether it pays off to perform the verification and validation in an automated way (e.g. using Jenkins[5], a continuous integration server).

[3] http://subversion.tigris.org/
[4] http://gforge.org/gf/
[5] http://jenkins-ci.org/

3 EVALUATION PROCEDURES

3.1 OVERVIEW OF EVALUATION PROCEDURES

3.1.1 Security

One of the main blocks that needs to be evaluated thoroughly in the INTEGRATE framework is the security block, any security leak and/or vulnerability can be disastrous for the project. This evaluation will mainly focus on: security, performance, interoperability, compatibility and user-friendliness. To describe the evaluation procedures of the security block in more detail, we use the security use cases of deliverable 1.4 as starting point.

A short repetition; **UC.SEC.1, UC.SEC.2** and **UC.SEC.3** describe use cases that are specific to identity management while **UC.SEC.4** describes a use case for the INTEGRATE access control management. More specific, **UC.SEC.1** contains the steps that a USER should follow to successful sign in on the INTEGRATE platform. **UC.SEC.2** contains the reverse operation of **UC.SEC.1**, namely the steps that a USER should follow to successful logout from the platform. **UC.SEC.3** describes how a USER is registered on the platform. Finally **UC.SEC.4** defines how access rights are set to a USER on the INTEGRATE platform.

For both the access control management and identity management a test scenario is described. In this test scenario we want to evaluate the security components by:

- Calculating performance values (benchmarks) in order to compare the implemented security block with other existing solutions.
- Getting input from end-users about the user-friendliness and ease of use in form of surveys.
- Performing compatibility, functional, security and interoperability tests.

If one of the tests/surveys in the test scenario is evaluated negatively, corrective actions will be taken. A more detailed description (use case level) of the security evaluation procedures in the test scenario is listed in the next section (ordered by software quality type).

3.1.1.1 Software quality

According to ISO 9126[6] exposed in the first section, the quality model identifies six main quality characteristics. In order to validate each of the ISO characteristics, the different methods of evaluation are exposed in relation with each of the use cases from the security block:

[6] *ISO/IEC IS 9126, Software Product Evaluation – Quality Characteristics and Guidelines for Their Use*, Geneva: International Organization for Standardization.

1) **Functionality:**

- **[UC.SEC.1] Sign in USER:** tests will be defined in order to check if the implemented sign in functionality complies with the defined user and legal requirements. Next to this, tests should check for possible vulnerabilities and/or security leaks in the sign in component. Finally tests will check the interoperability capabilities of the sign in component with other components. The outcome of these tests can result in possible functional changes to the sign in component.

  [UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

- **[UC.SEC.3] Register USER:** tests will be defined in order to check if the implemented registering functionality complies with the defined user requirements. Next to this, tests should check for possible vulnerabilities and/or security leaks in the registration component. The results of these tests will be used to make possible functional changes to the register component.

  [UC.SEC.4] Set access rights: * *Same as [UC.SEC.1]*

2) **Reliability:**

- **[UC.SEC.1] Sign in USER:** To conduct a comprehensive review of the stability of the sign in component, the number of software failures caused during execution of the functional tests will be recorded. Once having calculated the total number of failures that arise during the execution of the tool, conclusions will be made about the quality of the software. In order to do this it will be attempted to execute the functional tests in different failure scenarios. These scenarios will investigate for example software errors of system itself, such as memory access errors in retrieving the stored data (access to the queries or the result sets stored in memory) or failures in the connection with the server resulting from falls in the internet network. Finally, the time will be calculated that the platform needs to be operative after a reset of the sign in component; in order to measure the recoverability of the sign in component.

  [UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

  [UC.SEC.3] Register USER: * *Same as [UC.SEC.1]*

  [UC.SEC.4] Set access rights: * *Same as [UC.SEC.1]*

3) **Usability:**

- **[UC.SEC.1] Sign in USER:** End-users will be requested to provide feedback regarding the sign in functionality by means of surveys. These surveys will contain questions that focus on satisfaction, user-friendliness and ease of use of the system. The results of the surveys will be used as feedback to the developers with ultimate goal to improve the general usability of the system.

This feedback will generally have a big impact on the user interface of the framework.

[UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

[UC.SEC.3] Register USER: * *Same as [UC.SEC.1]*

[UC.SEC.4] Set access rights: * *Same as [UC.SEC.1]*

4) Efficiency:

- **[UC.SEC.1] Sign in USER:** The objective is to evaluate the degree of optimization that is possible for the sign in system. For this, different benchmarks (resource use, time behavior, etc.) are defined to get a general view on the overall performance of the sign in system in isolation. The results of these tests will be compared to predefined expected results and results of already existing sign in systems. When the system in isolation is performing well, the benchmark tests can be repeated for the sign in component in the global INTEGRATE framework.

  [UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

  [UC.SEC.3] Register USER: * *Same as [UC.SEC.1]*

- **[UC.SEC.4] Set access rights: *** *Same as [UC.SEC.1].* Special care needs to be taken in order to optimize the PDP decision engine. It is expected that the decision engine will be the main bottleneck in the access rights system, especially when complex policies are used.

5) Maintainability:

- **[UC.SEC.1] Sign in USER:** One of the architectural visions of the INTEGRATE platform is to have loosely coupled components (well-defined interfaces). To achieve this architectural vision, the implemented interfaces between the sign in component and connecting components must be evaluated. If the results of this evaluation are satisfying, the maintainability of the system can be assured. Next to this, a logging mechanism should be implemented that makes it possible to easily identify the main cause of possible failure in the sign in component.

  [UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

  [UC.SEC.3] Register USER: * *Same as [UC.SEC.1]*

  [UC.SEC.4] Set access rights: * *Same as [UC.SEC.1]*

6) Portability:

- **[UC.SEC.1] Sign in USER:** Most important for the portability quality is that the sign in system is conform to industry standards. As already discussed in other deliverables, SAML is a good choose for exchanging sign in requests and responses. Test needs to be written to check if the sign in system is conform to the SAML specification/stack. Next to the conformance attribute, a sign in component should be easily interchangeable with another sign in component (loosely coupled vision as explained in the maintainability section).

  [UC.SEC.2] Sign out USER: * *Same as [UC.SEC.1]*

  [UC.SEC.3] Register USER: * *Same as [UC.SEC.1]*

- **[UC.SEC.4] Set access rights:** Most important for the portability quality is that the access rights system is conform to industry standards. As already discussed in other deliverables, XACML is a good choice for exchanging access rights requests and responses. Test needs to be written to check if the access rights system is conform to the XACML specification/stack. Next to the conformance attribute, an access rights component should be easily interchangeable with another access rights component (loosely coupled vision as explained in the maintainability section).

3.1.2 Informed Consent

As in the security block, the use cases of Deliverable 1.4 are used as starting point for the description of evaluation procedures of the informed consent block. The evaluation will focus on performance, interoperability, compatibility, user-friendliness and to lesser extend security.

Seven use cases were defined in concerning the informed consent functionality. In **UC.IC.1** the registration of a signed informed consent form was explained. **UC.IC.2** verifies if a signed informed consent meets the criteria parameters. An investigator can list in **UC.IC.3** all the registered signed informed consents of a particular patient. The withdrawal of a signed informed consent of a particular patient is explained in **UC.IC.4**. **UC.IC.5** describes the creation of a new informed consent configuration for a particular trial. The edit of informed consent configuration is explained in **UC.IC.6**. Finally **UC.IC.7** contains the steps that need to be followed to activate an informed consent configuration.

Similar to the test scenario of the security block we want to evaluate its components by:

- Calculating performance values (benchmarks) in order to compare the implemented informed consent block with other existing solutions
- Getting input from end-users about the user-friendliness and ease of use in form of surveys
- Performing compatibility, functional and interoperability tests

If one of the tests/surveys in the test scenario is evaluated negatively, corrective actions will be taken. A more detailed description (use case level) of the inform consent evaluation procedures of the test scenario is listed in the next section (ordered by software quality type).

3.1.2.1 Software quality

According to ISO 9126 exposed in the first section, the quality model identifies six main quality characteristics. In order to validate each of the ISO characteristics, the different methods of evaluation are exposed in relation with each of the use cases from the informed consent block:

1) Functionality:

- **[UC.IC.1] Registration signed IC form:** tests will be defined in order to check if the implemented registration functionality complies with the defined user requirements. Other tests will check the interoperability capabilities of the registration component with other components. The results of these tests will, if needed, be used to make functional changes to the registration component.

  [UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

  [UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

  [UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

  [UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

2) Reliability:

- **[UC.IC.1] Registration signed IC form:** To conduct a comprehensive review of the stability of the registration component, the number of software failures caused during execution of the functional tests will be recorded. Once having calculated the total number of failures that arise during the execution of the tool, conclusions will be made about the quality of the software. In order to do this it will be attempted to execute the functional tests in different failure scenarios. These scenarios will investigate for example software errors of system itself, such as memory access errors in retrieving the stored data (access to the queries or the result sets stored in memory) or failures in the connection with the server resulting from falls in the internet network. Finally, the time will be calculated that the platform needs to be operative after a reset of the registration component; in order to measure the recoverability of this component.

[UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

[UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

[UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

[UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

[UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

[UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

3) Usability:

- **[UC.IC.1] Registration signed IC form:** End-users will be requested to provide feedback about the registration functionality by means of surveys. These surveys will contain questions that focus on satisfaction, user-friendliness and ease of use of the system. The results of the surveys will be used as feedback to the developers with ultimate goal to improve the general usability of the system.

  [UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

  [UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

  [UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

  [UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

4) Efficiency:

- **[UC.IC.1] Registration signed IC form:** The objective is to evaluate the degree of optimization that is possible for the registration system. For this, different benchmarks (resource use, time behavior, etc.) will be used to get a general view on the overall performance of the registration system in isolation. The results of these tests will be compared to predefined expected results and results of already existing registration systems. When the system in isolation is performing well, the benchmark tests can be repeated for the registration component in the global INTEGRATE framework.

  [UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

  [UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

  [UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

[UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

[UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

[UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

5) Maintainability:

- **[UC.IC.1] Registration signed IC form:** One of the architectural visions of the INTEGRATE platform is to have loosely coupled components (well-defined interfaces). To achieve this architectural vision, the implemented interfaces between the registration component and connecting components must be evaluated. If the results of this evaluation are satisfying, the maintainability of the system can be assured. A logging mechanism should be implemented that easily identifies the main cause of a failure in the registration component.

  [UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

  [UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

  [UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

  [UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

6) Portability:

- **[UC.IC.1] Registration signed IC form:** It should be tested that the registration component is easily interchangeable with another registration component (loosely coupled vision as explained in the maintainability section).

  [UC.IC.2] Verification signed IC form: * *Same as [UC.IC.1]*

  [UC.IC.3] List signed IC forms: * *Same as [UC.IC.1]*

  [UC.IC.4] Withdraw IC form: * *Same as [UC.IC.1]*

  [UC.IC.5] Create new IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.6] Edit IC configuration: * *Same as [UC.IC.1]*

  [UC.IC.7] Activate IC configuration: * *Same as [UC.IC.1]*

3.1.3 Trial Management

There are two trial management related use cases: *registering a trial to the INTEGRATE platform* (UC.TRIALMGT.1) and *Edit a trial of the INTEGRATE platform* (UC.TRIALMGT.2).

At this point in time, the trial management use cases are not elaborated yet. The use cases cover the management (e.g. creation, modification) of the metadata pertaining to the clinical trials in INTEGRATE. The trial management related use cases are secondary use cases; they exist to enable functionality such that the primary use cases (where the added value of the platform is offered to the end users) can be performed (such as the use cases of the screening scenario).

The two use cases are tightly coupled with the definition of the trial metadata model. This model describes all the details of the trials managed by the INTEGRATE platform, such as trial name, trial eligibility criteria, trial participants and so on.

3.1.3.1 Software quality

According to ISO 9126 exposed in the first section, the quality model identifies six main quality characteristics. In order to validate each of the ISO characteristics, the different methods of evaluation are exposed in relation with each of the use cases from the trial management block:

1) Functionality:

- **[UC.TRIALMGT.1] Registering a trial to the INTEGRATE platform:** Tests will be defined to verify that the implemented functionality complies with the user requirements. Though not yet defined in the requirements, it should be assessed on regular time points of the project whether interoperability is a necessity for trial management. If interoperability is deemed necessary, scenarios will be described in Deliverable 6.3 covering the interoperability[7]. The tests will verify that only authorized users can register a trial.

  [UC.TRIALMGT.2] Edit a trial of the INTEGRATE platform: * *Same as [UC.TRIALMGT.1]*

2) Reliability:

Reliability is of a lesser concern for trial management. These use cases are not performed in a (time) critical situation and do not face directly an end-users (clinical) workflow.

[7] At the moment of writing, an internal discussion is ongoing whether BRIDG is relevant w.r.t. interoperability.

- **[UC.TRIALMGT.1] Registering a trial to the INTEGRATE platform:** In order to assess reliability, the number of software failures encountered during the execution of the functional tests will be recorded. If the number exceeds a specified number, an investigation will be made with as goal to find solutions to improve the quality of the software.

  [UC.TRIALMGT.2] Edit a trial of the INTEGRATE platform: * *Same as [UC.TRIALMGT.1]*

3) Usability:

Trial management is performed by only a limited number of (expert) users. Of the usability properties, understandability is very relevant (to avoid mistakes in the metadata definition of clinical trials).

- **[UC.TRIALMGT.1] Registering a trial to the INTEGRATE platform:** The users will be surveyed to assess that the logical concepts conform to the view of the user.

  [UC.TRIALMGT.2] Edit a trial of the INTEGRATE platform: * *Same as [UC.TRIALMGT.1]*

4) Efficiency:

Due to the sporadic use and none (time) critical nature of the use cases, efficiency is not important for trial management. In addition, no excessive resource usage is expected.

- **[UC.TRIALMGT.1] Registering a trial to the INTEGRATE platform:** The functional tests will be annotated with timings which must not be exceeded while registering a trial to the platform. In addition, resource specifications (e.g. required hardware resources) will be defined which should not be exceeded.

  [UC.TRIALMGT.2] Edit a trial of the INTEGRATE platform: * *Same as [UC.TRIALMGT.1]*

5) Maintainability:

As INTEGRATE is a research project, the components should be conductive to change.

- **[UC.TRIALMGT.1] Registering a trial to the INTEGRATE platform:** The main item which is likely to change is the clinical trial metadata model. Metrics will be defined to assess the effort to cope with these changes.

  [UC.TRIALMGT.2] Edit a trial of the INTEGRATE platform: * *Same as [UC.TRIALMGT.1]*

6) Portability:

There are no requirements with respect to portability.

3.1.4 Screening Scenario

The main use case defined for the screening scenario is *UC.1 - Patient trial screening*. UC.1 requires a patient registered in the INTEGRATE platform. This functionality is described in *UC.2 - Registering a patient to the INTEGRATE platform*.

3.1.4.1 Software quality

According to ISO 9126 exposed in the first section, the quality model identifies six main quality characteristics. In order to validate each of the ISO characteristics, the different methods of evaluation are exposed in relation with each of the use cases from the screening block:

1) Functionality:

- **[UC.1] Patient trial screening:** Tests will be defined to verify that the implemented functionality complies with the user requirements. This scenario will have stringent requirements on enforcement of authentication and authorization as data of a patient is used to check eligibility clinical trials.

  [UC.2] Registering a patient to the INTEGRATE platform: * *Same as [UC.1]*

2) Reliability:

As INTEGRATE is a research project (as opposed to a product development project), a higher frequency of failure by faults in the software is acceptable. However, D6.3 will constrain the frequency.

- **[UC.1] Patient trial screening:** In order to assess reliability, the number of software failures encountered during the execution of the functional tests will be recorded. If the number exceeds a specified number, an investigation will be made with as outcome solutions to improve the quality of the software.

  [UC.2] Registering a patient to the INTEGRATE platform: * *Same as [UC.1]*

3) **Usability:**

- **[UC.1] Patient trial screening:** The users will be surveyed to assess that the developed software is usable. The survey will focus on satisfaction, user-friendliness and ease of use of the system.

  [UC.2] Registering a patient to the INTEGRATE platform: * *Same as [UC.1]*

4) **Efficiency:**

- **[UC.1] Patient trial screening:** Patient trial screening should be very efficient, and various metrics will be defined and measured. The results of the benchmarks will be compared to the current way of working to assess the improvement made by the INTEGRATE platform.

- **[UC.2] Registering a patient to the INTEGRATE platform:** Benchmarks will be defined to evaluate the minimum requirements w.r.t. resource usage and time behavior.

5) **Maintainability:**

As INTEGRATE is a research project, the components should be conductive to change.

- **[UC.1] Patient trial screening:** The amount of effort of implementing likely (but still unplanned) changes to the system will be assessed.

  [UC.2] Registering a patient to the INTEGRATE platform: * *Same as [UC.1]*

6) **Portability:**

There are no requirements with respect to portability.

3.1.5 Cohort selection

Six use cases are related to the cohort selection's evaluation procedure. ***UC.3*** can be considered the center piece. ***UC.3*** describes the definition, by the user, of a specific result set, the extraction of data items in which the user is interested, and the final retrieve of such data items from the platform to the user.

In order to determine a patient cohort, the use case ***UC.4*** described the specific criteria that the patient must fulfill. Query definition will allow the user to define a result set with ***UC.5,*** by applying, to all patients or to an already existing result set, a previously defined query or set of queries.

User may use to options to retrieve data from a result set: (i) with the use case **UC.24**, where the user selects the specific concepts from the core data set and obtains the corresponding data, and (ii) with the use case **UC.25**, where the user selects raw data objects. Both uses cases are derived from, and included in, the use case **UC.3**.

Finally, the alternative use case ***UC.6***, allows the user to display concepts form the common information model (CIM) that are directly related to a specific result set.

The designed test scenario will check:

- Correct definition of a set of queries
- Creation of a group of result sets derived from the previous queries
- Correct application of such defined result sets to a known group of patients
- Performance to select concepts to extract
- Correct retrieval of relevant data related to selected concepts
- If the obtained results agree to the previously known and expected results
- Correct storing of data by the platform
- The alternative scenario that allows the user to display CIM elements and concepts related to a determined result set

Finally, it will be also required to evaluate the response time of the platform and its efficiency and ease of use.

3.1.5.1 Software quality

According to ISO 9126[8] exposed at the first section, the quality model identifies six main quality characteristics. Next, in order to validate each of the ISO characteristics, such different methods of evaluation are exposed in relation with each of the use cases from the cohort selection:

1) Functionality:

- **[UC.4] Define a query:** A group of tests will be defined with the objective of creating a set of queries and checking their inclusion on the list of available queries. Such tests will also check correct removal and editing of the previously defined queries in order ensure correct functionality.

  [UC.5] Define a result set: * *Same as [UC.4]*

- **[UC.3] Obtain data for a result set:** A set of test with known results will be defined to check the correct retrieval of data from such results sets.

  [UC.24] Obtain CIM based data for a result set: * *Same as [UC.3]*

[8] *ISO/IEC IS 9126, Software Product Evaluation – Quality Characteristics and Guidelines for Their Use*, Geneva: International Organization for Standardization.

[UC.25] Obtain raw data for a result set: * *Same as [UC.24]*

- **[UC.6] Display CIM based data from a result set:** A set of tests will be defined to check the correct displaying of the data from the core dataset. Correct access and visualization of the information stored therein will also be tested.

2) Reliability:

- **[UC.4] Define a query:** To conduct a comprehensive review of the system reliability, a failure log will be implemented to store the number of failures during execution of functional tests. From software errors to system failures, such as memory access errors in retrieving the stored data and connection problems with server due to internet disconnection.

  [UC.5] Define a result set: * *Same as [UC.4]*

  [UC.3] Obtain data for a result set: * *Same as [UC.4]*

  [UC.25] Obtain raw data for a result set: * *Same as [UC.4]*

- **[UC.24] Obtain CIM based data for a result set:** UC.24 requires additional test due to the connection with the core dataset. Therefore, a log of possible failures arising from the interoperability with the core dataset will be also established.

  [UC.6] Display CIM based data from a result set: * *Same as [UC.24]*

3) Usability:

- **[UC.4] Define a query:** Satisfaction surveys will be designed to test the ease of use of the user interface. They will be answered by a group of users not related with the software development. These evaluations aim to evaluate the usability of the software[9][10] and to measure the quality of the developed interfaces. Questions that compose surveys will be defined according to the goals of the developments.

  [UC.5] Define a result set: * *Same as [UC.4]*

  [UC.6] Display CIM based data from a result set: * *Same as [UC.4]*

- **[UC.3] Obtain data for a result set:** *** *N/A due to the absence of interface*

- **[UC.24] Obtain CIM based data:** *** *N/A due to the absence of interface*

[9] Nielsen, J. *Designing Web Usability: The Practice of Simplicity.* New Riders.1991
[10] Norman, D. A. *The Design of Everyday Things.* The MIT Press. 1998

- **[UC.25] Obtain raw data:** **** N/A due to the absence of interface*

4) Efficiency:

- **[UC.4] Define a query**: The objective is to evaluate the degree of optimization of the platform with respect to the system. To do this, the time required to build a query according to the functionality of the use case. In addition, to measure the response time by the required functionalities, measurements of time required to connect with different services and system resources will be carried out.

  [UC.5] Define a result set: * *Same as [UC.4]*

- **[UC.3] Obtain data for a result set:** Performance test will be designed to test query execution and retrieval of data with massive data sources. For this purpose, real data is not required and a data generator can be used from a sample of real data.

  [UC.25] Obtain raw data for a result set: * *Same as [UC.3]*

- **[UC.24] Obtain CIM based data for a result set: *** *Same as [UC.4] including test of interconnection with core data set services*

  [UC.6] Display CIM based data from a result set: * *Same as [UC.24]*

5) Maintainability:

- **[UC.4] Define a query:** Considering that the architecture of the platform is loosely coupled, and that an agile approach has been selected for development, to ensure maintainability of the system related to use cases requires performing an analysis of the evolution of the developed software. This analysis will be used to estimate maintainability efforts for cohort selection use cases.

  [UC.5] Define a result set: * *Same as [UC.4]*

  [UC.3] Obtain data for a result set: * *Same as [UC.4]*

  [UC.24] Obtain CIM based data for a result set: * *Same as [UC.4]*

  [UC.25] Obtain raw data for a result set: * *Same as [UC.4]*

  [UC.6] Display CIM based data from a result set: * *Same as [UC.4]*

6) Portability:

There are no requirements with respect to portability.

3.1.6 Central pathology review

The Central Pathology Review Platform consists of seven use cases. The use cases are parts of the final scenario which was extracted from the stakeholders. The procedure of the central review process for a pathology image is in brief as follows:

In case that there is no Collaboration group, the administrator has to define a Collaboration group **UC.CR.1**. Then the administrator registers a "central review protocol" **UC.CR.3** which defines the images that have to be reviewed and also defines the responsible reviewers. Each of the reviewers is notified by the Central Pathology Review Platform if there are images which are pending to be reviewed. The review process **UC.CR.5** is a procedure in which the reviewer views the image, annotates it (if needed) and finally fills a report (which is submitted and stored in the platform). Finally when all the reviewers have finished their evaluations, the Central Pathology Review Platform checks their reports for possible conflicts. If there is a conflict the platform informs the reviewers and provides to them a resolution form with suggestions and possible solutions **UC.CR.6**.

3.1.6.1 Test scenarios

The central review process can be divided in two stages. The first stage is the one where the administrator assigns the reviewers to the images of interest (use cases **UC.CR.1** and **UC.CR.3**). The second stage is the review process carried out by the reviewers (use cases **UC.CR.5** and **UC.CR.6**). In order to evaluate the performance of the platform we define the two test scenarios, where each will handle the corresponding stage described above: A test scenario for the definition of the central review protocol which is carried out by the administrator and another test scenario for the review process, carried out by the reviewer.

The scenarios that will be checked are:

Scenario A

- The right creation of the collaboration groups.
- The proper management of the existing collaboration groups.
- The right definition of a new “central review protocol”.
- The proper management of the existing central review protocols (either pending or completed).

Scenario B

- The correct display of the pathology images.
- The smooth operation of the mechanism responsible for the annotations and reports.
- The proper handling of the images which were reviewed.
- The time to complete the operations by the user, and the ease of use of the platform in every stage of operation.

3.1.6.2 Software quality

1) Functionality:

For each of the following use cases a group of specific tests will be performed in order to ensure their proper functionality.

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** The objective of the tests performed for this use case is to successfully create a collaboration group. The tests will prevent the creation of duplicate groups and will also ensure that there is a proper submission of the data during the creation of a collaboration group.

- **[UC.CR.2] Edit a Collaboration Group of the "Central Review Platform":** The tests will check the proper editing and/or deletion of the already existing collaboration groups.

- **[UC.CR.3] Create/Define a new Task:** The tests will verify the proper registration of a "central review protocol" (more analytically, the assignment of images to the reviewers) by the administrator.

- **[UC.CR.4] Edit/modify a Task:** These tests will check the proper editing and/or deletion of a Task which has been already defined.

- **[UC.CR.5] Review and Annotation process:** The tests that will be performed in this case will check the smooth operation of the review process. In detail it will examine the accurate display of the pathology images, the fast navigation between the pathology images, the easy and friendly means of annotating the images and operation of filling the appropriate reports.

- **[UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts:** The tests defined here will check the proper display of the conflicts appearing in the reports of the reviewers regarding an image under review, and will help to extract useful results which will improve the appearance and the layout of the information, in such a way that the resolution of a conflict can be addressed quickly among the reviewers.

- **[UC.CR.7] History of the images that have been reviewed:** These tests will check the correct displaying of the images that have been reviewed.

2) Reliability:

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** To conduct a comprehensive review of the reliability of the software, the failures caused during the execution of the functional tests (of each of the use cases explained in the previous section) will be monitored, counted and logged. Conclusions about the quality of the software will be made based on the total number of failures arising from the execution of the tools Finally, it will be measured the time that the platform needs in order to become operative again, after a reset of all of its components; all with the aim of measuring the recoverability of the software.

  [UC.CR.2] Edit a Collaboration Group of the "Central Review Platform": ** Same as [UC.CR.1]*

  [UC.CR.3] Create/Define a new Task: ** Same as [UC.CR.1]*

  [UC.CR.4] Edit/modify a Task: ** Same as [UC.CR.1]*

  [UC.CR.5] Review and Annotation process: ** Same as [UC.CR.1]*

  [UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts: ** Same as [UC.CR.1]*

  [UC.CR.7] History of the images that have been reviewed: ** Same as [UC.CR.1]*

3) Usability:

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** A series of surveys of satisfaction and ease of use will be defined that will be answered by a group of users which firstly will not be related with the development of the platform and secondly will be among the group of the end users. These evaluations aim to evaluate the usability of the software and to measure the quality of the developed interfaces. The issues and questions that compound the surveys will be defined according to the goals and desires of the developers.

  [UC.CR.2] Edit a Collaboration Group of the "Central Review Platform": ** Same as [UC.CR.1]*

  [UC.CR.3] Create/Define a new Task: ** Same as [UC.CR.1]*

[UC.CR.4] Edit/modify a Task: * *Same as [UC.CR.1]*

[UC.CR.5] Review and Annotation process: * *Same as [UC.CR.1]*

[UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts: * *Same as [UC.CR.1]*

[UC.CR.7] History of the images that have been reviewed:
* *Same as [UC.CR.1]*

4) Efficiency:

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** The objective is to evaluate the degree of optimization of the platform with respect to the system. To do this, it will be measured the time required to carry out the execution of the functionality corresponding to each of the use cases. Once such time has been measured, it will be established if it is acceptable, compared to the expected execution time. The time measured will also be compared with the time required by other status quo current solutions. In addition to measuring the response time by the developed functionalities, the time needed to connect with the different services will be measured, and also the system resources spent to carry out with the execution of each of the use cases.

  [UC.CR.2] Edit a Collaboration Group of the "Central Review Platform":
  * *Same as [UC.CR.1]*

  [UC.CR.3] Create/Define a new Task: * *Same as [UC.CR.1]*

  [UC.CR.4] Edit/modify a Task: * *Same as [UC.CR.1]*

  [UC.CR.5] Review and Annotation process: * *Same as [UC.CR.1]*

  [UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts: * *Same as [UC.CR.1]*

  [UC.CR.7] History of the images that have been reviewed:
  * *Same as [UC.CR.1]*

5) Maintainability:

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** Considering that the architecture of the platform is considered loosely coupled and the development agile, the way to assure the maintainability of the use cases consists of establishing an analysis of the evolution of the developed software. This analysis will be based on the different changes to the software, on the times spent on its development, and on the difficulties found to carry out the modifications. Therefore, from such detailed analysis of the evolution of the platform, it will be identified and determined the efforts needed to make changes in the software, as well as the expected effects derived from hypothetic changes in the system.

- **[UC.CR.2] Edit a Collaboration Group of the "Central Review Platform":**
  ** Same as [UC.CR.1]*

  [UC.CR.3] Create/Define a new Task: ** Same as [UC.CR.1]*

  [UC.CR.4] Edit/modify a Task: ** Same as [UC.CR.1]*

  [UC.CR.5] Review and Annotation process: ** Same as [UC.CR.1]*

  [UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts: ** Same as [UC.CR.1]*

  [UC.CR.7] History of the images that have been reviewed:
  ** Same as [UC.CR.1]*

6) Portability:

- **[UC.CR.1] Registration of a Collaboration Group to the "Central Review Platform":** The Central Review platform for Pathology images has the following advantage, it is built upon liferay portal, which is a true portable java based web platform. Therefore once installed, the system would be available through a web browser, and thus automatically is characterized as a true portable application. To test the portability of the tool, it will be executed from all the major clients (Mozilla Firefox, Google Chrome, etc). The objective is to evaluate both the system performance, for each of the different combinations, like checking if the generated results are identical. For this evaluation, it will be run a series of basic tests and it will be collated if the generated results and the resulting log files are identical in all cases.

- **[UC.CR.2] Edit a Collaboration Group of the "Central Review Platform":**
  ** Same as [UC.CR.1]*

  [UC.CR.3] Create/Define a new Task: ** Same as [UC.CR.1]*

  [UC.CR.4] Edit/modify a Task: ** Same as [UC.CR.1]*

  [UC.CR.5] Review and Annotation process: ** Same as [UC.CR.1]*

  [UC.CR.6] Comparison of the images which were reviewed and Resolution of potential Conflicts: ** Same as [UC.CR.1]*

  [UC.CR.7] History of the images that have been reviewed:
  ** Same as [UC.CR.1]*

3.1.7 Analytical tools

The INTEGRATE Analysis Platform provides a framework with applications for the analytical tools for the statistical analysis of a cohort, and the sharing of predictive models for cancer prognosis and treatment response. The use cases related to the platform are mainly divided into three main categories; those responsible for the implementation of the statistical analysis (**UC.IAT.1** and **UC.IAT.2**), use cases **UC.PM.1** and **UC.PM.2** linked to the predictive analysis study, and the general use cases (**UC.IAT_PM.1**, **UC.IAT_PM.2**, **UC.IAT_PM.3**, **UC.IAT_PM.4**, **UC.IAT_PM.5**) which play a complementary role for publishing a new tool or model to the platform, and for visualization, download and storage of the analysis reports.

Briefly mentioned, with use cases **UC.IAT.1** and **UC.PM.1** the user selects a specific statistical analysis tool or a predictive model, related to a research question as defined in D.1.2 and D.5.1, respectively. Use cases **UC.IAT.12** and **UC.PM.2** serve to the interaction between the central data warehouse and the platform leading to the cohort's retrieval. With use cases **UC.IAT_PM.1**, **UC.IAT_PM.2**, **UC.IAT_PM.3**, **UC.IAT_PM.4**, **UC.IAT_PM.5** the user can access the analysis report generated by a selected tool or model at that time or previously stored to the platform, download the report, store the report to the platform, and get access to upload a new tool or model to the tools and models repository.

3.1.7.1 Test Scenario

The INTEGRATE Analysis Platform is the main end-user platform in which a user accesses a pool of available tools and models for the analysis of cohorts in a user-friendly manner. It exposes functionalities in which a user:

- Is confirmed as a validate user to enter the platform.
- Selects either to perform a statistical or a predictive modeling analysis.

- Is being directed to the UI components, related to the selected analysis scenario:

    1. The user gets access and can filter the retrieved, by the data-warehouse, data for analysis and create new cohorts.
    2. Selects a specific statistical tool or predictive model linked to a pre-defined research question.
    3. Performs the analysis by simply pressing the execution button of a tool or a model.
    4. The platform automatically generates the analysis report and makes it available to the user to download, display, and store it to the platform.

- The user gets access to reports previously stored on the platform.
- The user proceeds to a new analysis or disconnects from the platform.

In order to evaluate and confirm that the platform fulfills the six characteristics of the quality assurance, a multi-stage test scenario has to be defined. The designed test scenario will assess:

- The correct display of the retrieved data.
- If the data can be easily handled by the user (i.e. filtering the population and the variables of the data, creates new cohorts for analysis, etc.).
- The interoperability between the platform and the components that are hidden under the UI (storage place of the platform, the tools and models repository, the central data-warehouse, and the meta-data ware-house as defined in D.2.4).
- The computational complexity and processing time required to perform the statistical and predictive analysis.
- The correct storing of an analysis report and its meta-data information (date and time of creation, user who performed the analysis, etc.) to the platform.
- The ability of a user to download, displays, and stores the analysis reports generated by the tools and models.
- The adaptability and changeability of the platform to adopt and make usable new tools and models to the users.
- The sustainability, usability and stable multi-use of the platform.

3.1.7.2 Software Quality

1) Functionality:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** A number of several tests will be deployed. Each of these tests will have as objective to examine:

    - The proper functionality of the platform when a user selects a specific statistical tool (the user is driven to the platform's UI which is designed for the selected tool, the UI buttons are activated and the user can proceed to the analysis)

- The multi-user environment of the platform. The web-based architecture of the platform should have the capability of allowing more than one user at the same time to select the available tools for analysis.

[UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: ** Same as [UC.IAT.1]. In this case, the user(s) select(s) a model for predictive analysis.*

[UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: ** Same as in [UC.IAT.2]. In this case, the user(s) retrieve(s) and filter(s) the cohort that feed the predictive model.*

- **[UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse:** Tests with known results will be defined in order to check the data retrieval and interoperability between the platform and the central data warehouse. The INTEGRATE Analysis Platform, allows the visualization and filtering of the retrieved data to a new cohort. Therefore, tests will evaluate that the retrieved data are further filtered and passed to the tools for analysis correctly (i.e. selection of a specific clinical parameter for analysis, patients with age under a threshold, etc.).

- **[UC.IAT_PM.1] Visualization of the analysis report:** Several tests will check the correct returning of the results from a statistical analysis tool or a predictive model that has been run by the user in real-time. These tests will ensure that the correct analysis report, related to a specific tool or model is returned to the user.

- **[UC.IAT_PM.2] Visualization of the analysis report, stored to the platform:** Tests will evaluate the effectiveness of the INTEGRATE Analysis Platform in storing and retrieving all the reports that have been previously created by both the statistical and predictive analysis tools and models.

- **[UC.IAT_PM.3] Download of the analysis report:** Tests will check the availability of the platform in allowing a user to download the analysis report to his/her computer. Meta-data information related to the date and time of analysis, the user who perform the analysis, and etc. should be included to the report.

- **[UC.IAT_PM.4] Storage of the analysis report to the Integrate platform:** Several users will log-in to the platform, conduct a statistical or predictive analysis study, and store the analysis report to the platform.

- **[UC.IAT_PM.5] Publishing a new analytical tool or a predictive model:** Tests will assess the functionality of the platform to allow a user upload a new tool or model, and check that the platform's security controls accept a new tool or model to be part of the analysis platform.

2) Reliability:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** Several tests will be performed under a multi-user execution framework in order to:
    - Count any possible errors occurred during the interaction between the platform and the components that belong to its architecture (central data-warehouse, tools and models repository, etc.).
    - The successful completion of a statistical or predictive analysis using the platform's tools and models.
    - Evaluate the effectiveness of the platform's recoverability to bring back a failed system during the use of the platform, and the time required to establish a fully operative working environment after a failure.

- **[UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse:** A testing procedure will be performed under a multi-user environment, ensuring that:
    - No errors occurred during the interaction of the analysis platform with the Data Warehouse (i.e. connectivity issues).
    - The retrieval of the studied cohort is successfully completed and passed to the platform.

  [UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: ** Same as in [UC.IAT.1]*

  [UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: ** Same as in [UC.IAT.2]*

- **[UC.IAT_PM.1] Visualization of the analysis report:** A testing procedure will be performed to:
    - Evaluate failures in accessing, storing, and downloading the analysis reports generated by selected analysis scenarios.

  [UC.IAT_PM.2] Visualization of the analysis report, stored to the platform: ** Same as in [UC.IAT_PM.1]*

  [UC.IAT_PM.3] Download of the analysis report: ** Same as in [UC.IAT_PM.1]*

  [UC.IAT_PM.4] Storage of the analysis report to the Integrate platform: ** Same as in [UC.IAT_PM.1]*

  [UC.IAT_PM.5] Publishing a new analytical tool or a predictive model: ** Same as in [UC.IAT.1]*

3) Usability:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** The overall design of the INTEGRATE Analysis Platform architecture aims to administer a fully operational and easy to use platform for analysis of the clinical, genomic and imaging data. A complicated multi-functional framework is hidden under the platform's UI, consisting of procedures for executing the software for the statistical and prediction analysis, for retrieving the data from the central warehouse, the filtering of the cohort within platform, the execution of automatically generated analysis reports, and extra functionality for allowing access to the reports. A series of surveys of satisfaction and ease of use need to be answered by a group of users not related with these complex procedures. The surveys aim to confirm that the platform offers a fully-operational and a user-friendly manner in which even a user with no IT background can performs an analysis. The issues and questions that compound the surveys will be defined according to the goals and desires of the developers.

  [UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

  [UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: * *Same as in [UC.IAT.1].*

  [UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.1] Visualization of the analysis report: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.2] Visualization of the analysis report, stored to the platform: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.3] Download of the analysis report: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.4] Storage of the analysis report to the Integrate platform: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.5] Publishing a new analytical tool or a predictive model: * *Same as in [UC.IAT.1]*

4) Efficiency:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** The objective is to evaluate the degree of optimization of the analysis platform with respect to the system. Once the platform has been established, several tests will assess the computational effort and time need to execute all the available statistical analysis tools and predictive models. Several cohorts of different dimensionality will be created to "stress out" and act as a benchmarking test for the platform. Additionally, the computational time needed for the completion of an analysis through the platform, will be

compared to the optimal time needed for an analysis to be performed locally using the functionality running under the platform's UI.

[UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

[UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: * *Same as in [UC.IAT.1]*

[UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.1] Visualization of the analysis report: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.2] Visualization of the analysis report, stored to the platform: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.3] Download of the analysis report: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.4] Storage of the analysis report to the Integrate platform: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.5] Publishing a new analytical tool or a predictive model: * *Same as in [UC.IAT.1]*

5) Maintainability:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** Tests will be made in order to access and estimate the time and effort need to make a change in the statistical and predictive analysis software (i.e. develop or optimize software functions), a modification to the platform's architecture, as well as the expected effects derived from hypothetic changes in the system. Through the testing procedure, any possible failures in the software and the platform will be identified and avoided in the future.

  [UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

  [UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: * *Same as in [UC.IAT.1]*

  [UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.1] Visualization of the analysis report: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.2] Visualization of the analysis report, stored to the platform: * *Same as in [UC.IAT.1]*

  [UC.IAT_PM.3] Download of the analysis report: * *Same as in [UC.IAT.1]*

[UC.IAT_PM.4] Storage of the analysis report to the Integrate platform:
** Same as in [UC.IAT.1]*

[UC.IAT_PM.5] Publishing a new analytical tool or a predictive model:
** Same as in [UC.IAT.1]*

6) Portability:

- **[UC.IAT.1] Selection of an Analytical Tool from the INTEGRATE Analysis Platform:** To test the portability of the INTEGRATE Analysis Platform, all the components of the platform's architecture will be tested on multiple operating systems (Windows and Linux). From the client's point of view, the portability of the platform will be tested using multiple users and multiple browsers (i.e. internet explorer, firefox, safari, etc.)

  [UC.IAT.2] Selection of the examined patient cohort(s) from the Data Warehouse: ** Same as in [UC.IAT.1]*

  [UC.PM.1] Selection of a Predictive Model from the INTEGRATE Analysis Platform: ** Same as in [UC.IAT.1]*

  [UC.PM.2] Selection of the examined patient cohort(s) from the Data Warehouse: ** Same as in [UC.IAT.1]*

  [UC.IAT_PM.1] Visualization of the analysis report: ** Same as in [UC.IAT.1]*

  [UC.IAT_PM.2] Visualization of the analysis report, stored to the platform:
  ** Same as in [UC.IAT.1]*

  [UC.IAT_PM.3] Download of the analysis report: ** Same as in [UC.IAT.1]*

  [UC.IAT_PM.4] Storage of the analysis report to the Integrate platform:
  ** Same as in [UC.IAT.1]*

  [UC.IAT_PM.5] Publishing a new analytical tool or a predictive model:
  ** Same as in [UC.IAT.1]. Additionally, tests have to ensure that a tool or model that has been created under a different from the established system environment (i.e. platform is installed in UNIX and the user creates a tool in Windows)*

4 CONCLUSIONS

This document have detailed the procedures that have been designed to evaluate the INTEGRATE environment. First, the general quality assurance process for the evaluation of the project has been defined. A process based on general principles of software quality according to ISO 9126-1. Six different properties have been identified within the set of evaluation procedures to assure the quality of the INTEGRATE software solution: (1) Functionality, (2) Reliability, (3) Usability, (4) Efficiency, (5) Maintainability and (6) Portability. The main characteristics of the INTEGRATE platform have been then presented, emphasizing the different environments of the platform as well as repository requirements and software verification and validation services.

Once the proposed approach for the quality process was defined, the seven functionality blocks were detailed, according to use cases previously defined at D1.4. Specific evaluation procedures have been designed for each quality software property and for each use case. Therefore, each use case belonging to each block has specified a different “roadmap” of procedures and tasks, coordinated to assure the global quality of the proposed implementation.

5 REFERENCES

[1] *ISO/IEC IS 9126, Software Product Evaluation – Quality Characteristics and Guidelines for Their Use*, Geneva: International Organization for Standardization. 1991.