Evaluation criteria and verification procedures of the p-medicine platform

Project Number: FP7-ICT-2009-6 - ICT -6-5.3 - Virtual Physiological Human

Proposal: 270089

Deliverable id: D 15.1

Deliverable name: Evaluation criteria and verification procedures of the P-medicine platform

Submission Date: 31/01/2012

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	***p-medicine***
Project Full Name:	From data sharing and integration via VPH models to personalized medicine
Document No:	D 15.1
Document name:	Evaluation criteria and verification procedures of the p-medicine platform
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	1.3
Submission date:	31/01/2012
Editor: Organisation: Email:	Simona Rossi SIB simona.rossi@isb-sib.ch

ABSTRACT:

This deliverable,subdivided into two major parts, provides a list of evaluation and validation procedures for the p-medicine infrastructure.

1. Part 1: it provides recommendations for the development of quality procedures to ensure the quality of software in each technical work package;
2. Part 2: it provides evaluation criteria from the perspective of the end-user. It is focused on the ability of users to use the p-medicine infrastructure to build and run practical scenarios. Note: since the scenarios are tightly bound to the data architecture currently accepted, they may and will evolve with the architecture of the system.

The access to software management tools made available in the context of p-medicine is also described at the end of the present document.

KEYWORD LIST: Evaluation, validation, end-user scenarios, software management and quality assurance.

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nº 270089.
The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0	08/12/11	Draft	Simona Rossi
1.1	10/01/12	Draft	Simona Rossi
1.3	30/01/12	pre-final	Simona Rossi
2	31/01/12	Final	Simona Rossi

List of Contributors

- Simona Rossi, SIB
- Norbert Graf, USAAR
- Holger Stenzhorn, USAAR
- Marie-Luise Christ-Neumann, FhG-IAIS
- Elias Neri, Custodix
- Giorgos Zacharioudakis, FORTH
- Danny Burke, ecancer
- Ulf Schwarz, IFOMIS
- Alberto Anguita, UPM
- Emilio Sanfilippo, IFOMIS

Table of Contents

Executive Summary

Based on the available standards, evaluation procedures, scenarios and guidelines related to the evaluation and validation and to the software quality assurance (QA) of the p-medicine platform are provided in this document. Based on user needs and requirements (as defined in D2.02 and on patients' empowerment tools (D2.05), practical relevant scenarios for end-users (clinicians, biomedical researchers, data manager, trial manager and patients) have been proposed.

Their aim is twofold:

1. to translate the end-user requirements into practical features to be implemented in the p-medicine environment
2. to provide explicit steps for the evaluation and validation of that environment.

Given the finite resources available in the project, several scenarios have been selected to be implemented with priority compared to others.

The scenarios summarized in this document are covering all critical aspects of the deployment and anticipated usage of the p-medicine environment, from the management of software components, to the use of the entire infrastructure in the framework of clinical trials, data analysis and users information query and data retrieve (with particular attention to the patients as users in this scenario).

Software management tools used by the p-medicine developer community (or soon to be) are also described at the end of the document. These include a software repository, a framework for software validation through automatic testing and bug reporting tool.

Several template documents for QA reporting by WP leaders are provided:

1. list of the local evaluation managers
2. tool characterization table and ER schema
3. evaluation model, criteria and form
4. monitoring procedures

The implementation of those procedures for individual software components and scenarios will be available on the p-medicine server. The structure of the material presented in this document reflects the logical architecture of the p-medicine environment as it is envisioned at the time of writing and as was at the time of the project proposal. Adaptations of the procedures presented here will be needed, once the infrastructure architecture first schema and software components become available, to reflect their true functionalities and to remain of practical use in line with the p-medicine infrastructure evolution. Subsequent revised versions of this document will thus be issued in the course of the project to report on those adaptations.

This document represents also the base for the implementation of the certification as well as the evaluation of p-medicine tools within the ECRIN clinical trials infrastructure (p-medicine Task 6.2).

Introduction

With the huge presence of software systems and infrastructures in the modern society in general and in science, and people's increasing reliance on them in daily life, as well as a wide and variegated legislation addressing privacy and ethical issues, a rigorous approach for assuring that these software systems meet user's expectations for quality and reliability became a fundamental component of software production cycle. This is especially true for the p-medicine platform as this environment will be used to store and analyse real patient data, will host scientists running analyses in the stored data and will be used as a tool to help clinicians in their practice, providing visualization and additional information to help them refine their diagnoses and used by patients to be more confident about their disease, the treatment they could and they are receiving, exchange experiences and potentially build a more positive attitude toward their disase and life expectations.

In general terms, the quality expectations for software systems are two fold:

- the software must do the right things: software systems must do what they are supposed to do (end-user perspective)
- the software must do the things right: software systems must perform the tasks correctly (developer perspective)

These two aspects define two of the main components of the software quality assurance system (SQAS): the validation (does the software do the right things?) and verification (does the software do the things right?).

Accordingly, SQAS aims at ensuring a high quality of the software product through the related validation and verification activities. These activities must be carried out by the people and the organizations responsible for developing and supporting the system in an overall engineering process that includes:

- quality planning
- execution of selected quality assurance activities
- measurement and analysis to demonstrate software quality to all parties involved.

Unfortunately, as the complexity and code size of the software increase, the risks of having a failure increase as well, and there is no effective general solution to the size, complexity, quality and other software engineering problems. However, by following standardized software development practices and by addressing the quality issues during the whole life cycle of the software, the likelihood of such defects and the cost incurred by them (both to users and to producers) may be greatly reduced.

The purpose of this document is to propose the guidelines and a unified approach for ensuring the quality of the software products within the p-medicine infrastructure, in accordance with the user needs and requirements in D2.2, the patients' empowerment tools (D2.05) and the infrastructure architecture.

The implementation of this approach is adapted from various sources (ISO standards mainly) and involves both users and developers in the process of product testing. Due to the high complexity of the software to be produced and integrated in the p-medicine platform, this document does not attempt at covering all possible aspects of quality monitoring/ensuring for every module, but rather provide a template that should be adapted at the level of each module. As the various organizations involved in the software development process of p-medicine project have different internal QA policies/strategies, we reckon the need of a common approach to SQAS.

Two perspectives of quality evaluation are followed:

From a top-down perspective, various categories of end-users will evaluate the p-medicine platform in terms of its suitability to achieve its intended goals. The latter may be different for each category of end-user, thus the validation of the infrastructure is based on realistic scenarios for each of those categories. The scenarios list a series of anticipated results and intermediate goals to be achieved, which will allow measuring the performance of the platform in an objective way.

From a bottom-up perspective, technical work-packages develop software components which can be verified independently provided a range of boundary conditions i.e. sets of data for their interfaces. The modular organization of the p-medicine environment should facilitate the establishment of such verification procedures.

The evaluation of the p-medicine platform should clearly be viewed as an iterative process. Scenarios and QA procedures will evolve as new components get integrated in the environment or as some others are removed if considered useless. Revised versions of this document will thus be issued as p-medicine tools and end-user needs are refined and become available.

This document has the purpose to collect, describe and organize the material, methods and procedures that will support the deliverables 15.2 and 15.3 and more in general the following p-medicine objective: “Definition of scenarios and evaluation criteria supported by p-medicine”.

Local Managers

Being p-medicine a multiple aspects infrastructure, a good evaluation plan must be carried out in a concerted way by using use cases.

To carry out this issue in the most correct and complete manner we established local managers based on their p-medicine working areas among: data mining, user needs and requirements, data warehouse, data manager, patient empowerment, clinicians, semantic integration, infrastructure integration and quality.

Furthermore, the documentation provided here will constitute a documentation base for the evaluation of p-medicine tools within the ECRIN clinical trials infrastructure (WP6).

Table 1. Local Evaluation Managers.

Name	Institution	Area
Michael Mock	FhG-IAIS	Data Mining
Marie-Luise Christ-Neumann	FhG-IAIS	Usability/user needs and requirements
Benjamin Jefferys	UCL	Data warehouse
Holger Stenzhorn	USAAR	User needs and requirements/clinicians
Marian Taylor	UoXF	Clinicians/Data Managers
Danny Burke	ecancer	Patient empowerment
Alberto Anguita	UPM	Semantic integration
Giorgios Zacharioudakis	FORTH	Integration manager
Peter Coveney	UCL	Quality manager

Each local manager will run the evaluation process and send his feedback to the SIB, the integration manager and the quality manager. This will enhance the evaluation process modularity and a better understanding of the use cases based on the different areas. This partitioned iterative process will ensure the QA to be accurate and in line with the infrastructure and its components evaluation. Each local manager has the responsibility to run the evaluation process in his/her area of competence and report based on the results to the quality assurance work package manager (SIB) that will join and comment all the results received and give fedback to the local managers after discussion with quality manager, integration manager and good clinical practicioners.

The Figure 1 shows the quality assurance communication process.

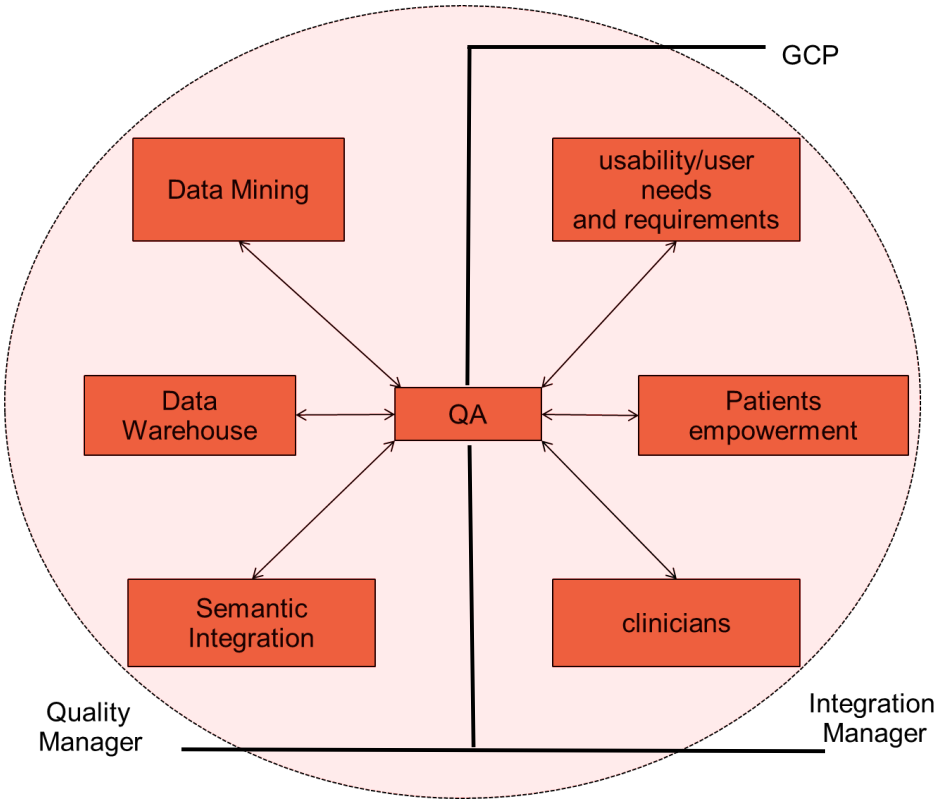


Figure 1: Iterative evaluation process organization schema. Evaluation procedures will be forwarded to the QA manager in an iterative way and a unified complete report will be sent to the quality manager, integration manager and GCP manager for knowledge and supervision giving feedback to the local managers at the end of the iterative step n.

Quality Assurance Guidelines

1 Introduction

A preliminary questionnaire has been conducted among all the partners, to list theirs internal quality guidelines. Based on the results and to the fact that already ACGT (http://www.eu-acgt.org/) used the ISO requirements, as well as those have been specified in WP2 to be used for usability purposes (part itself of the quality assurance plan), ISO norms will be used also for quality assurance.

The ISO (International organization for standardization, http://www.iso.org/iso/home.html) SQuaRE (Software product Quality Requirements and Evaluation), will be used as reference model; it lists standards in terms of:

- General Guidance: ISO/IEC 25000
- Particular Guidance: ISO/IEC 25040 (ISO/IEC 9126-1 and ISO/IEC 14598-1)
- Execution: ISO/IEC 25041 (ISO/IEC 14598-6), ISO/IEC 25042 (ISO/IEC 14598-3), ISO/IEC 25043 (ISO/IEC 14598-4)

When we discuss about quality evaluation we should keep in mind that we can view and run it from different point of view:

- Evaluation process for **developers** (ISO/IEC 14598-3), to use when the evaluation is conducted in parallel with the development
- Evaluation process for **acquirers** (ISO/IEC 14598-4), to use during modifications to existing software products

The general reference is contained in the evaluation **reference** model and guide (ISO/IEC 9126-1 and 14598-1) and structure and content of the **documentation** to be used is listed in (ISO/IEC 14598-6).

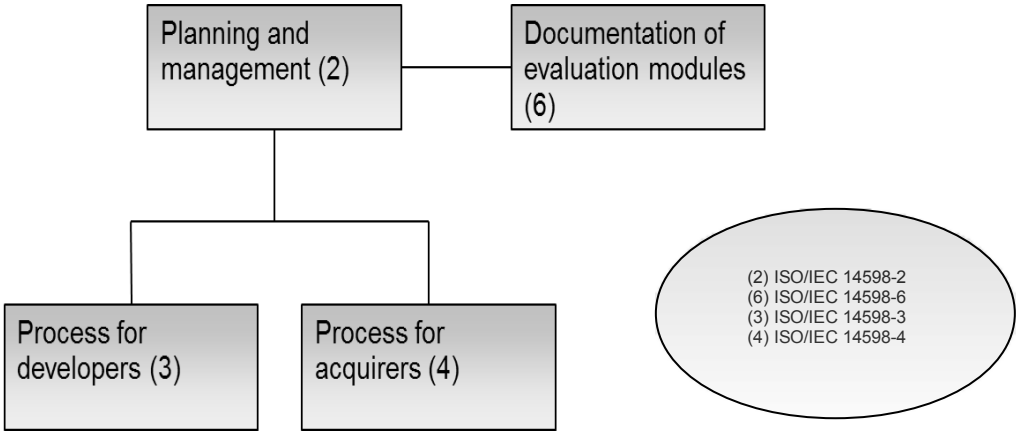


Figure 2. Adapted from ISO/IEC 14598 modules organization. The numbers in the round brackets represent the modules: i.e. if (2), then ISO/IEC 14598-2, etc.

2 Quality requirements categories

User needs specify also the required level of quality from the end user point of view. The defined requirements have to be seen from an external and an internal view as defined below:

- External SW quality requirements: are used as the target for technical verification and validation of the software product
- Internal SW quality requirements: quality from the internal view of the product, they are used to specify properties of intermediate software products. They may also be applied to deliverable, non-executable SW products such as documentation and manuals.

To assess quality levels the end user and/or evaluator have to receive a list of metrics that she/he can measure. Internal metrics are associated to the software product architecture and allow to predict the final product quality; external metrics are measurable when the product is under operation.

A scale must also have been defined; usually the scales can be divided into categories corresponding to different degrees of satisfaction of the requirements like:

- Exceeds requirements,
- target,
- minimally acceptable,
- Unacceptable.

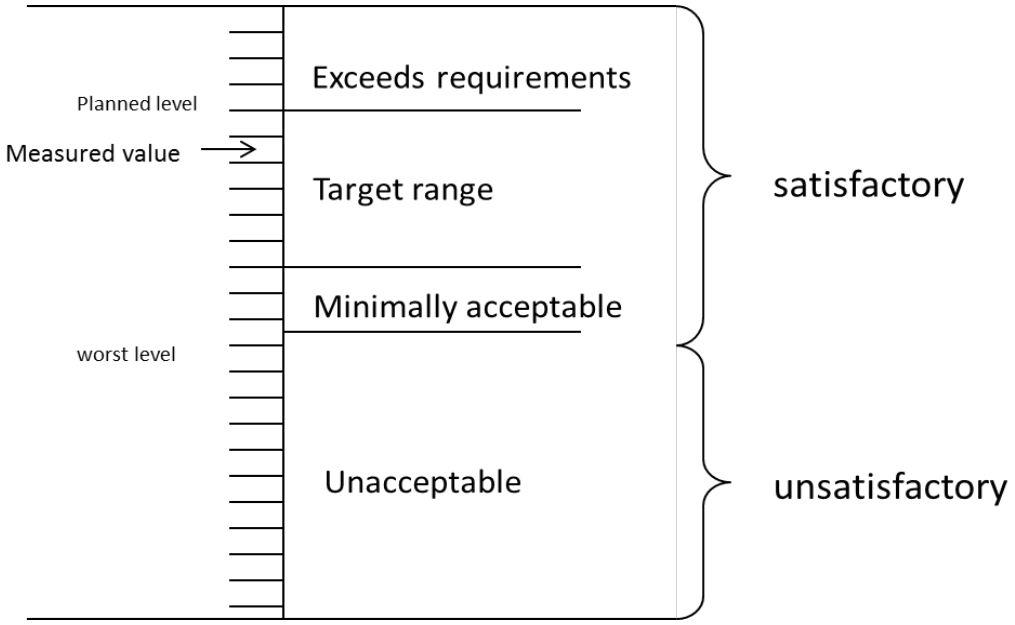


Figure 3. Adapted from ISO/IEC. Degrees of satisfaction and interpretation.

The categories should be specified so that both the user and the developer can avoid unnecessary cost and schedule overruns.

3 Quality model structure

A quality model is a set of requirements, entities and relationships that must be fullfilled to assess good quality.

The model should be structured in three main levels:

- Characteristic
- Sub-characteristic
- attribute

We can refer to two models of quality: the internal and external quality and the quality in use.

Internal and external quality

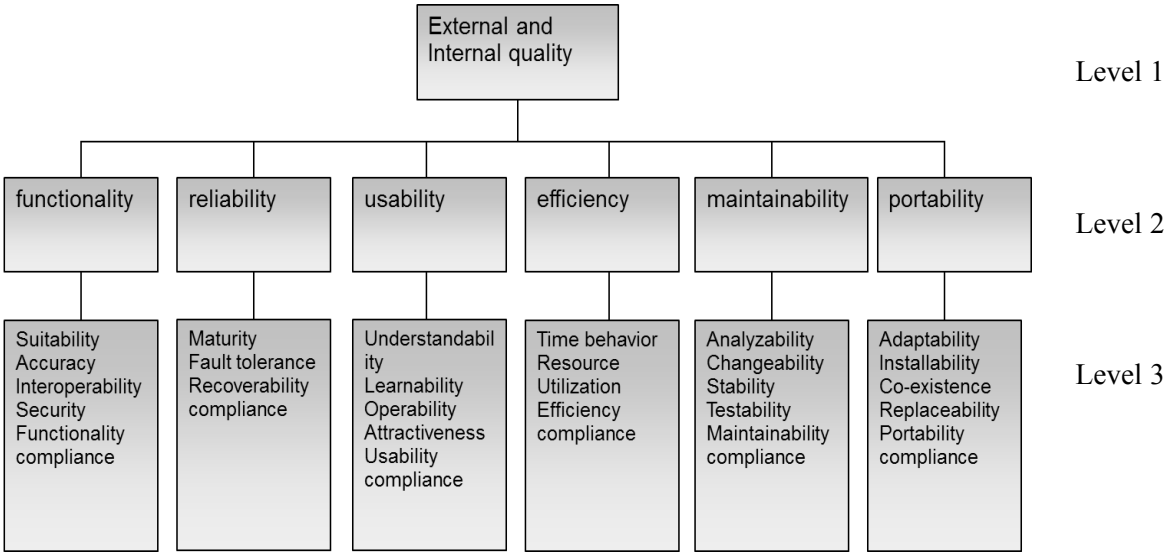


Figure 4. Adapted from ISO/IEC. Quality model for external and internal quality.

In this case we can see the main characteristics: functionality, reliability, usability, efficiency, maintainability and portability followed by the correspondent sub-characteristics.

The six listed characteristics are the fundamental aspects defined by ISO/IEC 9126 and can described as:

Functionality: a set of attributes related to a defined set of functions and specified properties. The functions are those that satisfy stated or implied needs. The sub-aspects include:

- suitability
- accuracy
- interoperability
- security

Reliability: those attributes pertaining to the capability of software to maintain its level of performance under stated conditions and for a stated period of time. The sub-aspects include:

- maturity
- fault tolerance
- recoverability

Usability: a set of attributes that bear of the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. The sub-characteristics include:

- understandability
- learnability
- operability

Efficiency: those attributes that bear on the relationship between the level of performance of the software and the amount of the resources used, under stated conditions. The sub-aspects include:

- time behaviour
- resource behavior

Maintainability: those attributes that bear on the effort needed to make specified modification. The sub-aspects include:

- analyzability
- changeability
- stability
- testability

Portability: a set of attributes conditioning the ability of the software to be transferred from one environment to another. The sub-aspects include:

- adaptability
- installability
- conformance
- replaceability

All these six aspects and the associated issues must be addressed both at the level of each module and the global level of p-medicine project.

Quality in use

Quality in use is the user's view of quality. Achieving quality in use is dependent on achieving the necessary external quality, which in turn is dependent on achieving the necessary internal quality.

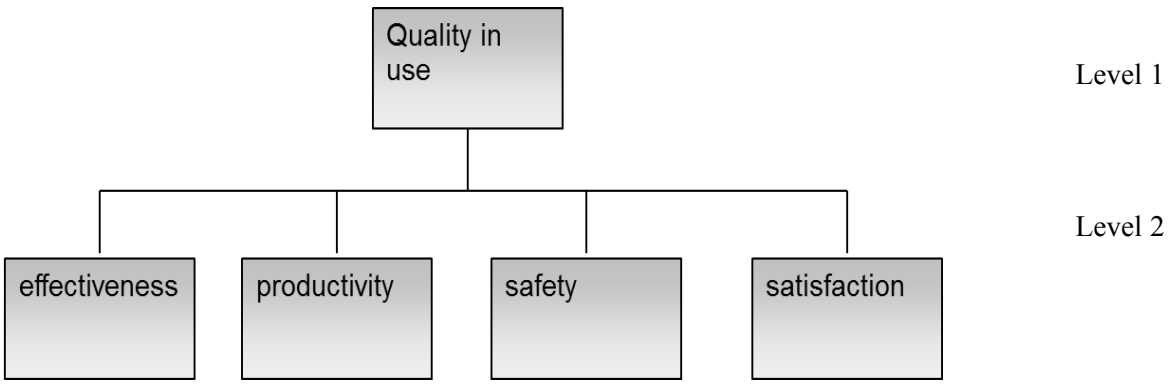


Figure 5. Adapted from ISO/IEC. Quality model for quality in use.

At level 2 we can define the four listed characteristics:

- **Effectiveness:** the capability of the software product to enable users to achieve specified goals with accuracy and completeness in a specified context of use
- **Productivity:** the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use
- **Safety:** the capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property or the environment in a specified context of use
- **Satisfaction:** the capability of the software product to satisfy users in a specified context of use.

4 End-user evaluation procedures

Introduction

End-user evaluation of the p-medicine infrastructure will be conducted through a number of realistic scenarios (D2.1) covering the anticipated usage of the infrastructure, from administration of the software components to specific clinical trials. The scenarios are based on the p-medicine scenarios described in Deliverable D2.2. For each step in the scenario, the required input data are enumerated and a description of the expected results is given. The steps listed for the execution of the scenarios respond to criteria which will help objectively rating the degree of success of the modules addressed therein.

The aim of each module is to test functional units of the p-medicine platform, such as the user interfaces. In detail, the user gets a realistic task that he/she has to conduct with the system, all interactions with the system are recorded by the tool named CamStudio[3]. From these records, evaluation reports are created where all weaknesses and real usage problems during execution of the user's task are identified and described (D2.2) together with the assessment of the quality of the offered tutorials. The modules are further subdivided into individual "atomic" tasks which form the actual units of evaluation. The success, partial failure or failure of a task is reported in a form provided for every module. In case of a failure, an assessment of its impact on the functionality of the tested component is performed.

Every module is described in terms of:

- Overview summarizing the functionality tested
- Required p-medicine tools (dependencies on other p-medicine components)
- Input data
- Expected results
- List of steps to achieve expected results
- Scenario evaluation form
- Assessment of rate of success (degree of satisfaction) and assign priority to uncompleted tasks (during evaluation workshops)

Levels of success available to report the outcome of evaluation steps:

- OK (target range of success): The task was accomplished as expected
- Partial (minimally acceptable): The task could be executed partially with minor loss of functionality
- Failure (unacceptable): The task could not be executed with an acceptable loss of functionality

Priority ranking for uncompleted tasks:

- High: describe tasks which incomplete state prevent the completion of the scenario
- Medium: describe tasks which lack non-critical functionality
- Low: describe tasks which lack some optional features

If applicable end-users will consider the following questions as basis for an overall assessment of usability evaluation of the tool under scrutiny:

[3] http://camstudio.org/

- Is the general interface suitable for your purposes?
- Rate the accessibility level (easy to use, hard, too complex)
- Is on-line help sufficient?
- Is the user manual well documented?
- Do you believe that additional training is necessary to apprehend the system?
- If yes, please precise on which functionality
- Are security mechanisms sufficient?
- Is the software free of errors that would make it possible to circumvent its security mechanisms?
- If errors occur, is the user able to understand the error message and can he/she eliminate the error within a conformable time?
- Are you satisfied with the personalization/customization features of the system?
- Is the quality of outputs/results acceptable?
- Are all parameters required by the program available?
- Are all inputs required by the program available?
- Are information processing delays acceptable: poor, fair, good
- Have you encountered any problem with the use of alphanumeric or special characters?
- To what degree is the p-medicine component interoperable with your existing IT environment/equipment? (poor, medium, high)

Organization of periodic evaluations, reporting

Most of the tests will be performed in a non-automatic way by the several classes of end-users[4][5].The advantage of automatic testing is that tests can be run frequently (e.g. daily), thus allowing a rapid discovery of problems related, for instance, to incompatibilities introduced when new versions of software components are installed. However, for scenarios requiring user-interaction, a human evaluation has to be performed. In p-medicine, this human evaluation process will be conducted in the context of workshops to be held in association with major milestones of the project, such as the delivery of demonstrators.

Various ways to organize the evaluation process exist, specifying for instance in detail the formal role of each actor involved in the process, however all share the following overall structure:

- Before the actual evaluation, documents have to be prepared that define which components have to be tested and specify the objective criteria to assess whether the evaluation is a success or not.
- During the evaluation process, the role of participants have to be defined, involving in particular a coordinator whose role is to ensure that the process is conducted in a proper way.
- After the evaluation process, the evaluation results are collected and summarized in an evaluation report.

[4] There are no automatic tests for usability pourposes.The test are conducted by the end-user with his task and therefore the tool CamsStudio is used as recording system. Only the interaction with the system will be recorded and evaluated by the usability engineer.

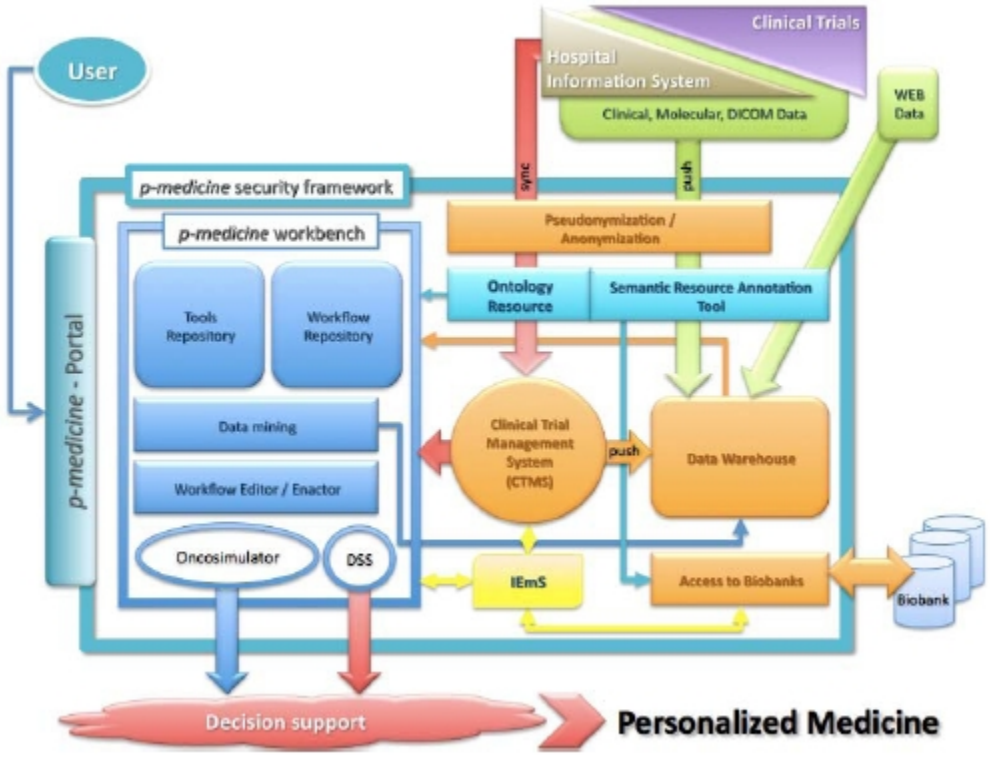
[5]For the semantic mediation tools, in the case of the Ontology Annotator it (web application with a graphical interface), this kind of tool might not be suitable for automatic testing. In the case of the Data Translation service: this service is accessed solely by the Data Warehouse whenever a PUSH operation is carried out. An automatic service that periodically pushes a sample database would allow testing both the PUSH service in the DW and the translation service.

During evaluation workshops, the actual testing should be performed as much as possible by representatives of the end-user community, although representatives of the developer community (p-medicine technical work packages), should also be present to provide support, clarifications and minor-bug-fixing capacity.

Reference p-medicine Architecture

For the time being, we refer to the architecture schema as in Figure 1.3.1-2 of the project proposal. Anticipated scenarios for the evaluation and validation depend heavily on the data flow inside the p-medicine infrastructure, which in turn depend on the architecture retained. The following figure describes the reference architecture underlying the present document.

Figure 1.3.1-2: The architecture of p-medicine from a clinical perspective



User needs and requirements

The ISO norms explain also how to take into account of the user needs and requirements by evaluating the stated and implied needs, the latter due to the fact that the users do not always reflect the real user needs because:

1. A user is often not aware of his real needs
2. Needs may change after they are stated
3. Different users may have different operating environments
4. It may be impossible to consult all the possible types of users

User needs and requirements have been identified in WP2 (D2.2). They give the basis for the development of the p-medicine environment used by the various user groups (bioinformaticians, biostatisticians, data managers, clinicians and patients).
The five context scenarios derived from the interviews, illustrate the special tasks with the whole context of use of the various end users. These tasks have to be taken into consideration with the usability standards ISO 9241 - part 11, 110 and 12 when developing the user interface with a common portal for all end users.

The user's task can be listed in a short summary. These summaries should be read carefully by the different developer groups concerning the work flow environment, the clinical and VPH tools. For a better understanding of the user's task it is necessary to read the whole context scenario of an end user of the special user group in the deliverable D2.2 Chapter 4 and D2.2 Appendix 2 Context Scenarios.

All listed tasks from the user's point of view in the following are without the derived system requirements and recommendations of the usability engineer. These system requirements concerning the seven dialogue principles are referred in the specific context scenarios in D2.2 Appendix 2.

1 Task of a bioinformatician

From the context scenario in D2.2 pages 126 – 135.

Bioinformaticians carry out statistical analyses on data extracted from high-throughput biological experiments" (from D2.2, page 126).

The analysis of a **bioinformatician** can be for example presented in the following steps:

1. read the raw data from "ncbi/geo" in the classical Affymetrix .cel format;
2. check the data quality with the tool R and make several plots and measures;
3. normalize the expression data, extracted by reading the .cel raw data files;
4. filter the Affymetrix probe sets based on the variance of the signal through the samples;
5. analyze the omics data in relation to the clinical-pathological variables in order e.g. to extract the genes differentially expressed between stage I and stage II samples.

The work is not mechanically, it differs from task to task.

Large omics datasets are analysed with related clinical-pathological variables like stage, age, gender, follow-up, etc.

R and Bioconductor are used as analyse tools as well as MatLab, Perl and C++ .

A simple workflow for analyzing Affymetrix expression arrays in R / BioConductor is described in the following steps:

1. loading the clinical data (load packages Affymetrix pre-processing and two-color pre-preprocessing; differential expression
2. import "phenotype" data: describing the experimental design
3. RMA[6] normalization and expression summary
4. identifying differentially expressed probe sets

Step 3 and step 4 are done in R / Bioconductor.

The workflow uses RMA from the affy package to pre-process Affymetrix arrays and the limma package[7] for assessing differential expression.

Clinical data is downloaded from:
http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM550623
and
ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SeriesMatrix/GSE22138/
to extract the complete table.

Raw expression data can be done directly in R, or by downloading the file GSE22138_RAW.rar from
http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE22138
to extract the complete table.

Results are usually saved as an rda file. The rda file is an R format to save data and variables. The user can directly load these files in R; in this way the data are ready to work with because they are already in the format needed to be input for the developed algorithms.

Results are reported to third parties or internally. The report is a combination of explanation of the analysis steps, intermediate and final results, visualized through plots, tables etc.

Usually the bioinformatician receives Excel files of the clinical data and the omics data are in the raw format, for example the .CEL for Affymetrix arrays (HG-U133 Plus 2.0, SNP6.0 etc.) or .txt for Agilent arrays.

To connect tasks, tools or components to make a clinical content table ready to be read in R will take few minutes till few hours depending on the special task.

The bioinformatician has to share the data because they work in groups and different aspects of the project, e.g. clinical data. This is common but the user works on one kind of data and a colleague works on another kind of data which belong to the same list of patient.

The project manager takes care of giving the direction of the project. He collects all results inside the group.

He/she uses a server for saving the code to reuse it and not losing time to do many things many times once more. He/she work with small components and merge these components at the end.

[6]http://rmaexpress.bmbolstad.com/
[7]http://bioconductor.org/packages/release/bioc/html/limma.html

The results are presented in R in form of plots.

Usually the bioinformatician makes changes to existing analyses workflows in using new available options or substitution of a step.

When creating eScience workflows or analysis processes the following phases exist:

1. Brainstorming
2. From brainstorming to entities and relationship definition
3. From entities and relationships to logical projects
4. From logical projects to physical projects
5. Evaluation by using scenarios / benchmarks

The scripts are reused. It happens pretty often that they have to exchange the code.

Changes can be made by updating functions, change parameters, insert new blocks and delete no longer used blocks.

Usually, tools are already selected, components are defined, much time is devoted to understand the task and how the components can be combined to satisfy the project's aim/s.

The bioinformatician uses the workflow top down to generalize the task. Usually he/she looks for the type of the project and thinks about the components. Then he/she tries to split the project into several components, e.g. tasks and faces. In his/her mind she thinks about a workflow.

The most time consuming task is the organization, to begin several steps and how to do them.
A stress factor for the bioinformatician is: doing the same kind of activity or doing many of the, like reading the manuscripts and learning a new language in a restricted time.

Wishes of the bioinformatician:

- to have a tool that automatic interactively check the input clinical data
- to be able working with Taverna for creating workflows
- to have a function, e.g. “abc” to create the user's own workflow with this function, or to have an icon on the screen; to have the opportunity to open a window in which the data can be imported
- repeated steps should be stored and performed again in the same way when necessary

2 Task of a biostatistician

From the context scenario in D2.2 page 149 – 154.

His/her work is data processing:

- data from databases, e.g. Access database, Excel files. The source are study databases of files created for the data of biological experiments;
- import data into his analysis system;
- analysis of the imported data;

- mistakes in the data have to be clarified. There are some difficulties which related to e.g. Excel specialities, for incompatibility of fields and data.
- missing procedure in SAS, he has to use R or to generate a procedure by himself

When there are the same analysis data for different projects, he writes similar programs in SAS to use them more than one time. He can re-use them.

Another task is:

- Planning of studies and
- analysing of biological data for presentations
- creating raw versions for the tables

Results are presented in form of:

- a paper,
- an administrator report,
- data monitoring in communities,
- just only input for a meeting or
- a presentation.

The analysis method has to be specified in advance for new prospective trials.
Interaction with the clinicians is needed for trial protocols.
When he gets data from different sources or from biologists, there is a lot of work to do to forward it so that he can use it.
The patient identification came from the lab and the people who are in the study database. This could be done automatically. If there could be a patient identification that is stored in each of the data sources they will get. This would be a good thing to have. Patient identification is very time-consuming. A kind of standardization would be fine.
He has to struggle with names which are the same but mentioned different things, codes for the same thing.

Technical support

He/she uses the Access Database or DBMS[8] from which he gets the input of clinical data. The most used analysis system is SAS statistical software (SAS is mostly used in the pharmacy industry). If there is no procedure in SAS, he has to use R or generates a procedure by himself.

3 Task of a data manager

From the context scenario in D2.2 page 136 – 148.

The data manager's responsibilities cover delivery of data management services to support clinical research, in particular:

- eCRFs for clinical trials of investigational medicinal projects
  and
- clinical databases associated with tissue biobank sample collections.

A certain amount of common work for each clinical trial is the following:

[8]Data Bases Management Systems

- review protocol,
- generate data requirements,
- development of eCRFs,
- testing,
- user and investigator acceptance,
- a period of up to a couple of years of prospective data capture,
- data cleaning,
- data lock,
- data export,
- linking with externally generated bio data.
- summary and analysis,
- data archive.

The data manager receives the data which varies according to the project:

- Trial data as indicated in the clinical trial protocol, typically
- Clinical trial patient demographics, previous medical history, previous treatments for the disease, informed consent, appropriate inclusion and exclusion data, trial drug treatments administered, results of physical examination, vital signs, haematology and biochemistry results, all concomitant medication, all adverse events experienced, data relevant to trial endpoints (e.g. response to treatment, duration of response, survival up to a certain time point), reason off study. Typically a lot of data items over a relatively short period of time, for a relatively small number of patients.
- Clinical site-specific databases supporting the biobank: typically less detail for a larger number of patients (thousands). Demographics, details of pathology and spread of tumour at first presentation (known prognostic factors), treatments given, outcomes, relapse/recurrence, survival data.

He/she should be able to generate a workflow by herself.

To produce meaningful complete and accurate datasets for collaborating research colleagues. They spend a lot of time capturing the data in a quality controlled way. Several colleagues could be access the same data set (e.g. trial eCRF database), but not the same record. Their working steps are fairly unautomated, therefore they have the flexibility (though possibly are therefore less efficient).

The one that she might be more involved in than most p-medicine partners is clinical data capture at the start. Clear data definitions and categorization.

Technical support

Clinical databases are developed using OpenClinica[9], open source and clinical trial specific software.
Site specific databases are developed in File Maker Pro or database software.
Both of these data collections have data sets exported into Excel, then pulled into Stata statistics package for analysis.
They use self-made components for statistical analysis, after the data has been (but the file of commands is not re-used) just re-created (generated).
Repository for new clinical trials, customizing designs used previously with most similarities, though there will be more customizing for different clinical databases as they are for different tumor sites.
Components depend on projects, in general:

[9]https://www.openclinica.com/

- Clinical trial modules/templates are re-used as a new separate trial eCRF is designed, export and summary steps are not currently re-used, just repeated in the new setting
- Site-specific databases: selection of export fields for analysis has been automated, with option to amend; statistical analysis steps are repeated to a certain extent

The data manager uses electronic clinical care systems, for which there is read only access. Data is re-typed into research databases. Also sets of patient notes used in clinic.

Common sets of instructions are not formally standardised within the data team, there are sets of work instructions.
It would be useful to have a sequence of producing a set of listings and tabulations for a clinical trial report plus a set of analyses relevant to the trial endpoints. Also routine listings that are required by law, e.g. annual safety reports for clinical trials, end of study report after closure.
The sequence is:

- select data set,
- export to Excel,
- load into Excel,
- carry out basis set of statistical summaries
- tests.

This is not standardized to run as one item. Would be useful for site-specific databases.
They run a lot of early phase trials on small numbers of patients, so their statistical analysis can be in the form of simple summaries.

Critical Incidents which already occurred

Regarding their clinical site-specific databases, which are used to provide diagnostic, treatment and outcome clinical data for use in statistical analysis of experimental data such as protein expression, gene expression.

- The Biobank requires ethical approval to collect samples and associated clinical data for those who have given informed consent.
    - When a set of tissue samples is required for a particular project, a steering committee decides whether to approve the project. Tissues are only released if approved. Similarly the accompanying clinical data is only released if approved.
    - Using either tissue samples or any clinical data should be strictly regulated, so there must be a balance between ease of access for bona fide uses and the risk of speculative use of a data repository that is easily available for unapproved reasons (even if they are good ideas).
    - Hence the linking, downloading merging and different data sets should be tracked; everything should maybe somehow have a project code attached, as well knowing the user.
    - Maybe that the process of exporting a snapshot of clinical data at a certain time point, for the purposes of analysis, should somehow be halted until some criteria are met, e.g. completing a form with details of approval by the steering committee?
    - In short it is important to maintain a balance between ease of access and appropriate ethically approved use.

- Generally clinical trials data is very tightly regulated, and also tissue bank related data is becoming more so. It might be prudent to apply the same standards to all types of data in the system.
- Currently most of our data has to be manually input. I think we are a long way off from having a live feed from the hospital clinical care systems. But there are negotiations around improving our efficiency by obtaining routine electronic downloads, then running a script to match and import relevant data items. This would apply to the clinical site-specific, long term, databases. An option to allow this would be helpful, the event that such downloads come to pass.
- When we provide clinical datasets for statistical analysis, they already have some additional coding and calculation done from the raw data, e.g. an oestrogen receptor score (value will be between 0 and 8, an integer), will be coded as positive with value>=3, negative otherwise.
    - Maybe some standard universally accepted code should always be include in the raw data capture modules; this would be more efficient: it would save adding that step into every task of data manipulation for analysis and, once validated once at source, will be less prone to error.
- There may be a risk that those relatively unqualified in the tasks may misinterpret easily producible statistical summary and analysis. For example could run a task to carry out an analysis that is inappropriate for the data set they selected (may be parametric tests on data that is not normally distributed), and draw false conclusions. Maybe annotations on steps that are carried out in a routine would help? (This test assumes the data is normally distributed and the dataset is greater than 20)
- There is a risk that those producing reports/listings do not fully appreciate the meaning of what they think they are asking for; for example, looking for patients who had a certain procedure versus looking for that procedure (one patient may have had many procedures)
- The user's database software does not work well for much statistical analysis.
- Excel is not great for producing survival curves.

4 Task of a clinician in different roles

From the context scenario in D2.2 page 155 - 164

Trial chairman

In the role of a **trial chairman** he has the following tasks:

1. administrative activities and compliance with legal regulations
2. defining new trials or applying these trials
3. designing trials, i.e. graphically combining single events such as to create a trial design.
4. verifying and validate patient data during trials
5. providing (patient) data input into trials
6. management of patient data, analysis of the data, publication of trial results and providing advice to participating centres on any questions relating to the trial

As a chairman and administrator, he/she is fully responsible for each trial that he created and he is in charge of. He is the only person entitled to assign and distribute the rights to the single trials.

As an administrator (trial chairman), he arranges the trial, including all the content data such as graphical elements (templates = Case report form (CRF)), and also determines what rights and roles will be assigned and who will have access to the trial.

In addition, he draws up the trial protocol specifying all the details of the trial.

When creating a trial, the trial chairman has to focus on questions such as

- What will be the objectives of the trial?
- What should the trial be like?
- Which is the content of the trial?
- How will it be organized?

The software should offer him support for this functionality so that he gets a guideline in form of a master protocol and can access already existing or create CRFs. A graphical implementation of the trial would be useful.

Moreover, the software should be modular and extensible so that the clinician can attach specific modules to the existing software, for example.

The clinician builds a clinical trial containing a module of a basic data set, as is the case in other trials.

This module is saved in a CRF form that can also be used in other trials. Then there is a module, for example, which sends DICOM (**D**igital **I**maging and **CO**mmunications in **M**edicine) files or a file for imaging which is used for trial A as well as for trial B and also for trial C. Therefore, the software should be designed in a modular way so that the clinician can select exactly what he needs in the current situation.

The trial chairman instructs the system which participating hospitals and patients will be involved in the trial.

The trial chairman is administrator only for those trials that are managed by himself/herself.

Trial participant

In the role of a **trial participant** he/she only conducts the following task:

- providing (patient) data input into trials.

A clinician should have the possibility to enter data into the system during the daily working process through RDE (Remote Data Entry). The CRFs (Case Report Forms) needed for this purpose should be easily retrievable and editable for the clinician. Sending the completed CRFs to the trial centre should also be an easy task.

A trial contains all patient data which are necessary for the treatment process. In this trial, the **treating physician** gave a full description of the diagnosis of the patient (child). The treatment methods and the appropriate medication are listed as well. Side effects, Severe adverse events (SAEs) and Suspected unexpected severe adverse reactions (SUSARs) caused by the medications are listed by the treating physician. The trial participant would use the new software to help him perform his task efficiently and satisfactorily. The reporting should be done automatically.

The **treating physician** is the only person who has access to the trial for which he himself has provided the patient-specific information. Drawing up the trial protocol should be an easy task based on templates in the system so that the physician will not need to think and care

about the actual state of the art regulations and standards. These tasks should be made available by the system automatically using a regularly updated master protocol.

The software should recognize already during the registration process which role and which functions the physician/clinician will subsequently perform. The administrator can dedicate the role and rights for new users by choosing from a predefined list or manual modifications.

The trial participant is interested to register patient data in a trial. These patient data include:

- age
- gender
- affliction
- earlier infections (previous medical history)
- genetic disorders in the family
- etc.

The completeness of patient data is implicitly essential in order not to distort the assessment and evaluation of the data and to enable the right decisions for further treatments.

Validation of data is essential and should be easily performed by the data manager.

An important feature is the interface of the patient trial. Given the large number of trials, the interface should always be the same in order to enable the physician to find the desired trial quickly and to use the same procedure without needing to think about it.

The software must deliver results to the physician in order to reduce his workload in the daily working process.

When the clinician is in a clinical trial he would like to be able to extract any data, e.g. a relevant treatment graph, and then set it on a "scratchboard," to import the questions from the statistic module and collect them on the "queryboard". Questions are created automatically and subsequently sent to the statisticians for analysis. The result is sent back to the clinician/physician who can visualize it in the form of a life table or a descriptive analysis, for example.

The visualization tool may also be a “stand-alone” tool that may be used by the statisticians for other purposes as well. The clinician or physician can use this tool to see the results of his registered data and will have a broader information base.

A doctor in a clinical trial wants to use the software to be guided through the trial. This has to happen intuitively and, if possible, self-descriptively.

For the clinician as a non-statistician it is important to only collect the data and to get the results after the analysis that he is interested in. He wants to work using one workflow only. The results that he wants to see as a life table or as a frequency distribution that can be analysed using the R-analysis tool, or the result is delivered in the form of a bar chart or list. The system should support all different forms of result visualization.

The clinician can even generate a report that can be used as a doctor’s letter specifying the entire therapy of the patient and including all data. This is helpful, as it saves the local doctor a lot of time.

The data are anonymized before being admitted into the database and before a trial will be chosen. The only person entitled to see the data is the person who registered the data, the chairman and persons with dedicated rights to see the data.

The local doctor can only display data from his own clinic. In a given trial, all data collected for this trial were available to the trial chairman.

When logging in into the system the physician is automatically assigned a specific role, whereas in another trial he may have a different role and other rights.

5 Task of a trial manager in a clinic

From the context scenario in D2.2 page 165 - 180

The task of a trial manager in a clinic can be concentrated in:

- IT based support of the conduct of clinical trials, especially quality aspects
- enabling clinical trial system interoperability by the use of data standards (e.g., CDISC, HL7, ISO)
- use of patient data from HIS and other care data for clinical research (secondary use)
- quality management of clinical trials (site audit, SOPs)
- electronic archiving of clinical trials documentation
- clinical trial system validation
- ECRIN data centre group: clinical data management of ECRIN
- IT support of the management of clinical trials.

The trial manager deals with the legal framework for clinical trials in the EU. This is important for international clinical studies which must consider the different regulations and ethical guidelines.

The trial manager supports the physicians at the University Hospital in the task of conducting clinical tests. For example, SOPs and templates (e.g. for the trial protocol, Informed consent, AE messages, etc.) are made available.

What happens when a clinical trial is created either by a sponsor or a leading investigator?

- the clinical trial is in both cases the basis of a research idea;
- the leading investigator is interested in a surgical procedure that is better than the standard procedure;
- the pharmaceutical company is interested in whether their new drug has a better effect or is safety than existing drugs;
- there have to be templates for the necessary content of a trial protocol (e.g. templates for the important consent form);
- for the approval of a clinical trial, a protocol including the informed consent form, insurance confirmation, approvals by the Ethics Commission and by the competent authorities is required;
- if it is an interventional trial of a medicinal product it must be conducted according to the GCP guideline;
- the planning of GCP trials will be supported by the Coordination Centre for Clinical Trials. This centre has e.g. templates for the necessary content of a trial protocol, templates for the informed consent form, cover letter to the Ethics Committee, to the authorities, etc.;

- trial protocol and informed consent form must be signed (this is a requirement of GCP);
- additionally, the trial physicians must demonstrate to have received appropriate training;
- the investigator and additional specialists can make comments on the trial protocol and modify the plan as required;
- after the trial protocol has been accepted and finalised, it will be submitted to the Ethics Committee and the competent authorities for approval.

In all these activities the trial manager has to be supported by the software in an efficient and effective way.

After the investigators (trial physicians) have been recruited and trained, the trial can start.

The search for suitable patient populations could be executed by search in a data warehouse or other database with anonymised patient data.

The next step is the recruitment of patients:

- the investigator (trial physician) checks the inclusion and exclusion criteria. There are inclusion and exclusion criteria that determine whether a patient can participate in the trial;
- the patient has to sign an informed consent form that may be stored in paper form, currently. The patient is informed from by the trial physician about the trial. This task of the investigator may not be delegated to any doctor or assistant;
- the physician has to provide the explanation and information in the language of the patient.

After the recruitment, the patient will be randomised:

- he/she is given a trial number and an assignment to a treatment arm;
- during informed consent the patient must have the opportunity to ask questions. The patient must have the opportunity to leave the trial without fear of negative consequences. The patient must also be informed that he/she always has that freedom of choice;
- if the inclusion and exclusion criteria are fulfilled, the patient has consented and the consent form is signed, then he can participate in the trial and he/she is randomised;
- at randomisation, there are sometimes two or three arms, e.g. one treatment arm with the new medicinal product and one arm with the standard drug (or standard treatment);
- randomisation is supported by use of a software tool. The investigator (trial physician) sends a fax that he/she has recruited one patient to obtain a randomisation number;
- with the delivered randomisation number the investigator (trial physician) knows which randomisation arm must be assigned to this number.
- then the corresponding treatment of the patient can start.
- there may be different visits, in which the patient will be analysed and / or treated, e.g. to get a drug or is irradiated;
- the data of the visits are recorded. Blood samples may be examined, for example the blood of a patient is sent to a central laboratory for further analysis;
- the data of the analysis is sent back to the investigator for input into the CRF. The requirement is that the laboratory is certified;
- the review of certification documents is also part of the monitors' task.

One problem of unrecognized side effects:

- Side effects, which the patient communicates to the investigator (trial physician) are documented by the doctor. But there are a number of adverse side effects that may be not directly associated with the drug or the treatment and therefore may be not taken into consideration.
- Severe adverse effects (SAE), which may result in death and other severe results, must be reported immediately.

For the management of serious adverse effects, there is special software in use, SafetyNet. The trial physician must evaluate each message of an adverse effect, if there is a connection with the treatment or not. The data of SAEs must be coded according to MedDRA. Ideally the safety management system and data management system are integrated. But these data-bases or systems are often not yet integrated. There is no common data dictionary.

It would be desirable that SafetyNet, MedDRA-coding and data / document management can communicate together. The vision would be to have a common data dictionary as basis for an integrated network.

Useful would be software that supports the user in the quality analysis in clinical trials.

For the recruitment of patients it is important to get an idea of which hospitals can recruit how many patients and what may be the number of drop-outs, etc. For example, are there clinics with a large number of patients who suffer from a specific cancer disease?

It would be desirable if the lead investigator could see, how many patients have been re-cruited in which centers and in which period were the patients recruited? In case a center has not been able to recruit the planned number of patients, the question is why have no more patients been recruited? Where is the problem?

In case the physician has entered incorrect data a query management system is turned on automatically. The system creates a query and asked the investigator for the correct data. This system should guarantee the correct entry of data.

After all patients have been treated, last patient out, then all data are in the database. The database is closed after a quality check has been performed (database lock).

After the database has been locked, the statistical analysis of patient data can start.

The patient has the right to see his/her data. Currently, only the investigator has access to the patient data of a trial, but not the patient him/herself.

One problem may be, that if a patient has a health problem years after the end of the trial, this data are normally not considered in the trial.

Another point is to import laboratory data directly into the data management system.

6 Patients' needs

Patients' needs and recommendations are described in D2.2 page 60

A cancer patient's most interest is to get more information about:

- her/his disease
- the opportunities to be involved in a clinical trial
- the quality of life after the suggested treatments
- the side effects of the suggested treatment
- the latest new treatments
- the treatment options available to the patient (in her/his particular circumstances)
- how effective the different suggested treatments are
- the survival rates of the suggested treatments
- the best questions to ask the doctor for most relevant information
- how often the hospital treats this disease
- how successful the patient's hospital has been treating patients like her/him
- how often her/his doctor treats this disease
- who are affected by the same illness

From the different background knowledge, profession and technical experience of the cancer patients it is absolutely necessary to develop the user interface in a very easy, comprehensible and self-descriptive way. In any situation the user has to know where he is and what will be the next step to get the expected information.

The dialogue principles of the ISO 9241 – 110 illustrate an approach to identify the most important usability aspects for the interaction of the user with the dialogue system. The adaptability of each principle depends on the user group and their context of use.

The dialogue should present only the information that is necessary for the user to conduct the task successfully. For the system means this to provide information concerning the disease or treatment in an easy and comprehensible way.

The registration process for getting more information about the own treatment and its progress should be easy and intuitive performed. It should be self-descriptive and controllable. All medical expressions should be explained in the user's language. The shortcuts should be also explained in a comprehensible way.

If the patient got side effects from medical products or from chemotherapy she would be interested to have the possibility to exchange experiences with other cancer patients. It could also be helpful for her emotional / psychological support. A forum would be a possible solution for this request where cancer patients could discuss their experiences, problems and treatment progress with other cancer patients in the same situation.

Cancer patients would like to inform about the competent rehabilitation centres. A list of competent centers could be presented and described so that the patient can chose the best one for her recovering. The system could present a list of specialists for the different kind of cancer as well as the best questions to ask the doctor for most relevant information.

A list of specialists for the different kind of cancer should be available via the system. For more detailed information read the whole text in D2.2 page 60 and the following.

Considering all end-user groups a common portal will be developed to enable clinicians, bioresearchers, data managers and at least patients to use the software for conducting their task and achieve their aim in an efficient, effective and satisfied way. This can only be realized when the requirements of all user groups are taken into consideration.

Evaluation process' life

The evaluation process can be described in phases as shown in the following figure:

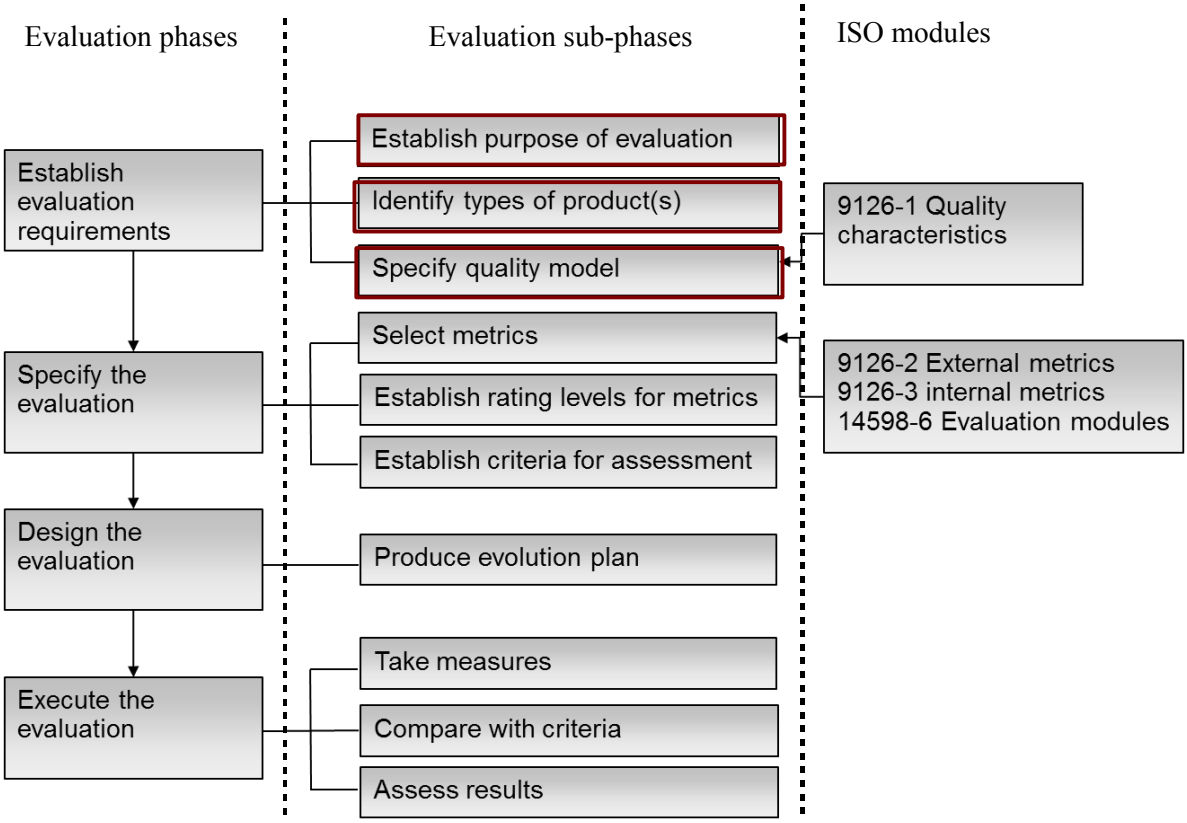


Figure 7. Evaluation process' life schema. Adapted from ISO/IEC.

The evaluation purpose, the structure of the evaluation model and how we would like to set up and its development are part of this document. ISO helps us with step-by-step content guidelines as specified in the following paragraphs.

1 Purpose of evaluation guidelines

To complete this task we must complete the following actions:

1. Decide on the completion of a process and when to send products to the next process
2. Predict or estimate end product quality
3. Collect information on intermediate products in order to control and manage the process
4. Decide when to release the product
5. Select a product from among alternative products
6. Assess both positive and negative effect of a product when it is used

2 Evaluation model structure guidelines

The model can be seen as the union and interconnection of 5 sub-modules plus an additional one that is nevertheless recommended:

1. EM0: provides formal information about the evaluation module and gives an introduction to the evaluation technique described in the evaluation module
2. EM1: defines the scope of applicability of the evaluation module
3. EM2: provides relevant references
4. EM3: includes the definitions needed for the evaluation module
5. EM4: specifies the input products required for the evaluation and defines the data to be collected and measures to be calculated
6. EM5: contains information about how to interpret measurement results
7. EMA: includes the detailed procedure for applying the evaluation module (recommended)

More in detail ISO defines the module' Table of Contents as follows:

3 Development of evaluation modules – guidelines

It requires at least 5 actions:

1. A.1 Identification of the evaluation module requirements
2. A.2 specification of the evaluation module
3. A.3 development of the evaluation procedure
4. A.4 description of the evaluation procedure
5. A.5 verification and validation of the evaluation module

Tool Characterization

Inspired by the VPH-NoE VPH ToolKit Guideline Document V1.01[10] a Tool Characterization form has been defined that help us in the quality assurance process, and in selecting the tools/functions/actions to be evaluated since not all of them can be evaluated.

The form to be filled is useful for tools already available that we want to use in p-medicine (having a well defined characterization helps the p-med partners to better understand and discuss together what the developers would like to use or re-use), and for tools we intend to develop (during the p-med life the stored information will help first deciding the tools to evaluate and to actually evaluate them).

Filling the attached form is a way to have a schema of technical elements and usability tools related, that will help focusing on pro and cons of every tool.

1 Tool Usability

To define the Tool Usability section, it is very important to consider the ISO 9241-11, 110 and 210. The ISO 9241-210 describes the human-centred design activities. Only when all developers have an explicit understanding of users' tasks and environments the design of the system can start. This implies that the designer has not only to look on the system from his/her point of view but also to recognize the seven dialogue principles, the task of the user with the whole context of use. Users have to be involved throughout design and development. The design is driven by user-centred evaluation. Therefore feedback from the user is a critical source of information.

"The requirements specification is refined iteratively by using scenarios early mock-ups and prototypes to obtain feedback from users on whether these incorporate the user requirements correctly and completely", taken from ISO 9241-210 Human-centred design for interactive systems.

After the requirements analysis has been finished with the common understanding the interaction design can start. This means to transform the key task into a dialogue design with guided information to the user.

A task or part of a task can be simulated or demonstrated by the system. Mock-ups could be helpful features to show the interaction of the user's task with the system.

The questions from the user's point of view are:

Which support comes from the system so that the user can conduct his/her work efficiently? Training, installation, recommendation, support, community...

What has the user to enter or to choose? Access, ability to join data, query search ability, easy to use…

Which information must be available firstly and which relationships must he/she recognize..., support, training, community

These are all questions the designer has to ask by himself when developing the system.

These questions are partially answered by the Tool Usability characterization form.

[10]http://toolkit.vph-noe.eu/toolkit-guidelines

Tool Characterization Form

	term description	***Tool - example: R***
Tool Information		
name	tool identification	R
version	version of the tool	2.14.0
function	action(s) performed by the tool	R is a free software environment for statistical computing and graphics, it provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.
specialty	domain of application	bioinformatics, statistics
I/O data	accepted and produced data and their formats	txt, csv, xls, rda, Rdata, jpg, png, pdf
I/O Data dimension	dimension	based on the available CPU/RAM
License	type of license (open-source or proprietary), if open-source specify the sub-type	open GNU
Certification	if yes, provide the name of the external organism that provided the agreement	http://www.r-project.org/ , then go to Certification
		-
Tool Specification		

Language	native programming language, compiler version	R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.
OS	OS where the tool is available	Windows, MAC OS X, Linux
Installation Recommendation	how to install the too? Required libraries? Drivers? Software?	http://www.r-project.org/, then go to Manuals, then go to R Installation and Administration
Third party libraries		http://stat.ethz.ch/CRAN/ and www.bioconductor.org
Type of tool	usually one among the categories: software, libraries, development environment, standalone application, web application, web service, framework,	software
Type of Computation	kind of computation? (CPU, GPU, DCI or HPC)	CPU
Tool Description		
short purpose	quickly explanation of the tool purpose	R is an integrated suite of software facilities for data manipulation, calculation and graphical display.
documentation link	any type of documentation (code documentation, tutorials, course, wiki, help system, mailing lists…)	http://www.r-project.org/, then go to Documentation

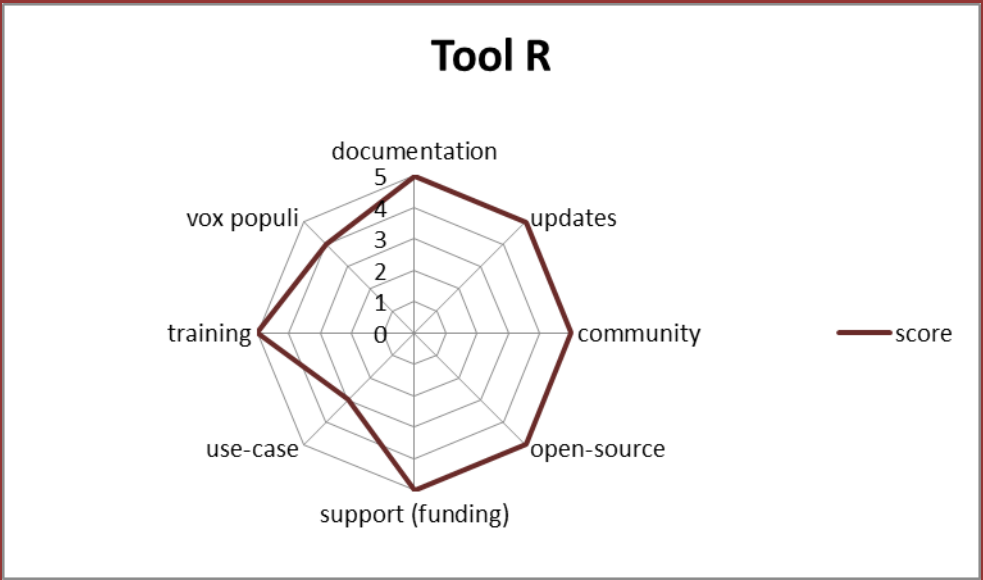
keywords		data manipulation, calculation and graphical display
citation & reference papers	paper(s) that describe the tool and paper(s) that use the tool	http://www.r-project.org/, then go to Books
shapshot(s)		http://www.r-project.org/, then go to screen shots
long purpose		R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes an effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis and display either on-screen or on hardcopy, and a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities. The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software. R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly. Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics. R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.
testing	tests performed on the tool, validation procedures and results	http://www.r-project.org/ , then go to Certification

download links	to the tool, the documentation, the libraries	http://www.r-project.org/
Tool Context		
people involvement	single person, team	http://www.r-project.org/, then go to Foundation and Member and Donors
author(s)		Robert Gentleman, Ross Ihaka
support		http://www.r-project.org/, mailing lists
how many people involved		world wide
reactivity	helpdesk, traceability system	http://www.r-project.org/, then go to R Project, then go to Mailing Lists, or Bug Tracking or Developer Page
type of collaboration	where the tool is coming from? What kind of collaboration? (european project, international initiative..etc.)	international initiative
funding status	institute, community behind the tool development	http://www.r-project.org/, then go to R Project, then go to Foundation and Members & Donors
institute/organization	software developers, researchers, clinicians, industrials…etc.	software developers and researchers
end-users target	future improvements, release to implement, planned new features (sustainability)	continuosly updated and founded worldwide
development plan	where more information are	main web-site: http://www.r-project.org/

	available	
website		http://www.r-project.org/
use-case		vignettes, like in http://www.bioconductor.org/packages/release/bioc/
training & course		http://www.r-project.org/, then go to Conferences
rights	rights to use, promote, upload, modify the tool	free
Tool Testing		
source of information		http://www.r-project.org/
software management tools		SVN (Subversion), https://svn.r-project.org/R-dev-web/trunk/index.html.
automatic testing		regression testing of R packages on multiple Oses
testing tools		make + diff on a set of scripts
testing frequency		daily
testing overview		http://cran.r-project.org/web/checks/check_summary.html
bug reporting		https://bugs.r-project.org/bugzilla3/index.cgi
scalability		
Tool Usability		

End-user categories	If it has a user interface, who are its potential users (IT experts, bioinformaticians, researchers, clinicians, patients, general public)	
scenario		
Access		
Ability to join data		
Query search ability		
Easy to use		
Interaction	If it is expected to interact with other tools/services, please name (or describe) the tools and the type of interaction (invocation, data exchange, produces/consumes data that are consumed/produced by the other tool etc)	
Security Tools Characterization - to check with Custodix		
number of credentials	yes/no	
adding users accounts	yes/no	

removing users accounts	yes/no	
setting permissions	yes/no	
acquiring credentials	yes/no	
easy to install	yes/no	
changes to security policy	yes/no	
Tool Ranking	each component must be evaluated and converted into one of the following levels: 1=poor, 2=poor-moderate,3= moderate, 4=moderate-good, 5=good	
documentation	evaluate the quality and quantity	5
updates	frequency	5
community	sustainability in general (mailing lists, FAQ pages, forums, workshops, etc.,)	5
open-source	participation to the development, bugs reporting, finding solutions	5

support (funding)	How many people are involved? How are they involved?	5
use-case	use-cases associated to the tool	3
training	Training course, workshop, online training?	5
vox populi	end-users belief	4
graphical visualization of tool ranking	score	

To facilitate the memorization, the visualization and the querying of the information stored in the forms, a database has been created. The ER below illustrates the entities and the relationships among them The IP issues are integrated; this might be especially important in case of tools that are built out of other tools. There is also a table dealing with the modularity of tools (called Modularity).

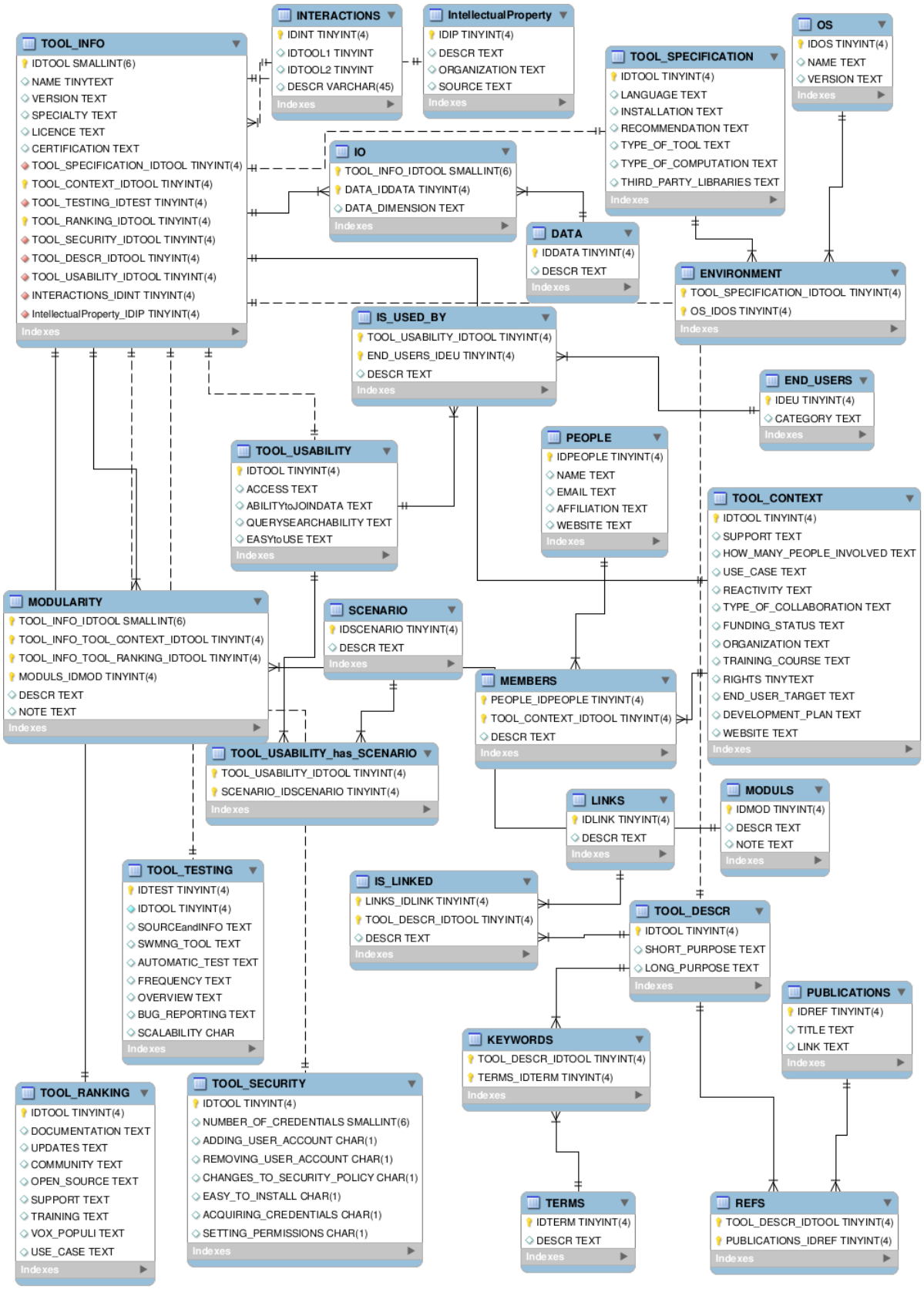


Figure 8. Tool characterization database: E-R schema.

List of tools in development

We here provide the list of tools under development at the time of writing this document. The list, continuosly updated has a quality role in the sense that improves clearness and communication inter- and intra- partners, avoiding duplication and possible lack of integration of the entire project as well as in all its parts.

1 *Ontology Annotator tool*

- **Name of the tool/service** : Ontology Annotator tool
- **Institution(s)** : UPM
- **Contact persons** : Alberto Anguita (aanguita@infomed.dia.fi.upm.es), Miguel García (mgarcía@infomed.dia.fi.upm.es)
- **Short description** : This tool is aimed at providing users with a GUI to annotate an existing database schema in terms of HDOT (the ontology to be developed in p-medicine for describing the data contained in the data warehouse). The results of this annotation process will allow to seamlessly access heterogeneous data in terms of the HDOT model, thus achieving semantic integration of such data. The tool will assist users in establishing semantic correspondences between an external database schema and the HDOT model.
    - **User profile** : this tool is intended for clinical database administrators/managers who wish to include their data in the p-medicine environment. The tool will, to some extent, hide the complexities associated to RDF/OWL models, making it suitable for users lacking this kind of knowledge.
    - **Type** : web application
    - **Input data** : schema of an external database (Excel, Access, SQL or RDF formats will be accepted)
    - **Output data** : XML document containing the annotation of the input schema in terms of the HDOT ontology
    - **Interaction with other tools** : the output data is submitted to the data warehouse

2 *Data Translation Service*

- **Name of the tool/service** : Data Translation service
- **Institution(s)** : UPM
- **Contact persons** : Alberto Anguita (aanguita@infomed.dia.fi.upm.es), Miguel García (mgarcía@infomed.dia.fi.upm.es)
- **Short description** : This service is aimed at translating data coming from the external (heterogeneous) sources into RDF triples matching the mapping schema (subsumed in the HDOT model). The DTS is to be invoked by the DW only. When the DTS receives a PUSH request from an external user to push external data into the DW, the DTS will carry out the required data translation operations to convert the external data into RDF triples matching the mapping HDOT-based schema. The translated data (in RDF format) is then forwarded to the Data Warehouse. The service will be deployed on the internet, as a web or REST service (to be decided in the future), in a secure fashion to ensure data confidentiality.
    - **User profile** : this service is to be accessed exclusively by the DW, as the service it offers is designed specifically for it.
    - **Type: web service**
    - **Input data** : data from external sources as pushed by an end user (data administrator/manager), annotation of the database involved

- **Output data** : translation of the data provided, in terms of the HDOT ontology (RDF format)
- **Interaction with other tools** : this service is employed solely by the DW

3 p-medicine portal

- Name of the tool/service : p-medicine portal
- Institution(s): FHG-IBMT, FORTH, USAAR, CUSTODIX, UCL
- Contact person: Fatima Schera, fatima.schera@ibmt.fraunhofer.de
- Short description: Web portal based on the open source Liferay portal framework .Tools and Services of p-medicine should be integrated into the portal as portlets or should be linked in the GUI of the portal.
- Potential users: clinicians, clinical trial leader, researchers, scientific community (mathematician, physicist, informatics, …), patients and their relatives, data managers (clinicians, nurses, documentalists) , technical support (portal and data base administrators, developers), representatives from legal and ethical committees, auditors
- Reference: http://www.liferay.com

4 Sync Services

- Name of the tool/service : Sync Services
- Institution(s): FHG-IBMT, USAAR, CUSTODIX
- Contact person: Gabriele Weiler, gabriele.weiler@ibmt.fraunhofer.de
- Short description: Services to retrieve data from hospital information systems for their reuse in ObTiMA. Data is retrieved from data repositories in which current data of HISs is stored, it is not yet clear if that are communication servers installed in hospitals or the data warehouse.
- Potential users: clinicians

5 Biobank access framework

- Institution(s) : FHG-IBMT
- Contact Person: Stephan Kiefer, stephan.kiefer@ibmt.fraunhofer.de
- The biobank access framework will be developed as a set of loosely coupled services .

6 p-BioSPRE - The p-medicine Biomaterial Search and Project Request Engine

- Contact Person: Christina Schröder, Christina.schroeder@ibmt.fraunhofer.de
- Short description: p-BioSPRE is a metabiobank that provides researchers the possibility to search for and request biomaterial. p-BioSPRE will be based on the CRIP metabiobank concept (http://www.crip.fraunhofer.de/) including a web application. Its integration into the p-medicine portal is foreseen.
- Potential users: Researchers

7 p-BioBank Wrappers

- Contact Person: Christina Schröder, Christina.schroeder@ibmt.fraunhofer.de
- Short description: A p-Biobank Wrapper enables a user to upload his biomaterial and related data in p-BioSPRE. Technically a p-Biobank Wrapper is based on the Integrative Research Database from the CRIP toolbox. It is a local server that is installed at the site of a biomaterial owner and configured to link one or more of his biobank management systems, in which data is stored that he wants to share. The p-Biobank Wrapper provides an interface to support a biomaterial owner to import and manage sample data from the linked biobank management systems. The process is realized by an import service that pseudonymizes the imported data according to the p-medicine concept via a trust center. The p-Biobank Wrapper comprises push services, which enable a biobank owner to push selected biomaterial data into the data warehouse.
- Potential users: Biobank owners

8 ObTiMA Trial Biomaterial Manager

- Contact Person: Stephan Kiefer, stephan.kiefer@ibmt.fraunhofer.de, Gabriele Weiler, gabriele.weiler@ibmt.fraunhofer.de
- Short description : The trial biomaterial manager will be developed as a component of the web based trial management system ObTiMA to enable management of biomaterial data in clinical trials and sharing selected biomaterial data. The trial biomaterial manager will provide an import service that enables users to import excel files with existing biomaterial data. Biobank data can be uploaded to p-BioSPRE (upload services provided). The trial biomaterial manager/ObTiMA integrates push services that are able to push selected biomaterial data into the data warehouse.
- Potential users: clinicians, biobank owners

9 CATS

- Name of the tool/service : CATS
- Institution(s) : Custodix
- Contact person : Elias Neri, elias.neri@custodix.com
- Short description : CATS is a service-based de-identification solution. CATS supports anonymisation of different types of data (XML, DICOM, Text, Database, ...) in a generic and extendable way.
- Potential users : data providers and data managers.

10 PIMS

- Name of the tool/service : PIMS
- Institution(s) : Custodix
- Contact person : Elias Neri, elias.neri@custodix.com
- Short description : PIMS combines a Master Patient Index (MPI) service with functionality for securely managing patient identities (and biological material) in clinical research environments.
- Potential users : data providers and data managers.
- Reference : PDF flyer (available on request)

11 Security Framework

- Name of the tool/service : Security framework
- Institution(s) : Custodix
- Contact person : Elias Neri, elias.neri@custodix.com
- Short description : At the moment it's not yet clear what the security framework will encompass. Possible security components are an Identity Provider, Integration Libraries, a Security Gateway (for authentication and authorisation), a User Enrolment and Management Site (a web frontend and possibly REST Services) an Authorisation Policy Generation Site or Tool (where administrators can configure authorisation rules), a Central Policy Decision Point, ...

12 Oncosimulator

- Name: Oncosimulator (Web integrator of OS-BRCA, OS-WT, OS-ALL)
- Contact Persons: F. Misichroni (faymisi@central.ntua.gr), D.Dionysiou (dimdio@esd.ece.ntua.gr), G.Stamatakos (gestam@central.ntua.gr)
- Type: Web Application
- Interaction: This version of the oncosimulator will be incorporated into the p-medicine portal.
    - Probable interaction with the Workflows using RESTful Web Services.
    - Potential interaction with the TUMOR repository (models repository), using RESTful Web Services, in order to retrieve the update version of the corresponding oncosimulator (breast cancer, nephroblastoma or acute lymphoblastic leukemia).
    - Potential use of the computational resources from the cloud in order to execute the simulations.

- Institution: ICCS

- Input data: files (CSV files, image files, xml files or other type of files) command line values, etc

- Output data: files (CSV files, image files, xml files, log files, zip files or other type of files)

- Potential users of the user interface: bioinformaticians, researchers, clinicians

- References: More information can be found in the deliverable D.12.1 “Architecture and information flow diagrams of the Oncosimulator and the biomechanism models”.

OS-BRCA

- Name: OS-BRCA (The Breast Cancer branch of the Oncosimulator)
- Contact Persons: K.Argyri (kargyri@mail.ntua.gr), E.Kolokotroni (ekolok@mail.ntua.gr), D.Dionysiou (dimdio@esd.ece.ntua.gr), G.Stamatakos (gestam@central.ntua.gr)
- Type: Standalone Application written in C++.

OS-WT

- Name: OS-WT (The Wilms Tumor (Nephroblastoma) branch of the Oncosimulator)
- Contact Persons: E.Georgiadi (egeorg@central.ntua.gr), D.Dionysiou (dimdio@esd.ece.ntua.gr), G.Stamatakos (gestam@central.ntua.gr)
- Type: Standalone Application written in C++.

OS-ALL

- Name: OS-ALL (The Acute Lymphoblastic Leukemia branch of the Onsosimulator)
- Contact Persons: E.Kolokotroni (ekolok@mail.ntua.gr), E.Ouzounoglou(elouzou@central.ntua.gr), D.Dionysiou (dimdio@esd.ece.ntua.gr), G.Stamatakos (gestam@central.ntua.gr)
- Type: Standalone Application written in C++.

13 Biomechanism Models

- **Name**: Molecular-level cancer simulator
- **Institution**: UCL (University College London)
- **Contact person**: Shunzhou Wan (shunzhou.wan@ucl.ac.uk)
- **Short description** :

The molecular-level cancer simulator is currently a standalone application that performs molecular dynamics simulations across multiple supercomputing resources. The simulator is possible to be plugged into the p-medicine Oncosimulator model at a later stage. The simulator will be incorporated into the p-medicine software architecture for its rapid and automatic execution. The simulator could be used to study the activation mechanism of key proteins, drug-protein interactions, and mutation effects on the activation and interactions. The simulator has potential to identify drug treatments better suited to an individual's specific genotypes. It can therefore be expected to have an increasing impact in personalized drug treatment of targeted therapy. The potential users would be bioinformaticians and researchers, and clinicians at a later stage to assist clinical decision-making. The structure of drug-protein complex will be the main input file for the simulator. The files can be obtained from the protein data bank (PDB, http://www.rcsb.org) if the structures are available for the complex, or using docking methods if only protein structures exist, or using homology modelling method based on a related homologous protein. The mutation status is another input, which indicates individual molecular characteristics of the tumour. Simulations will be performed on high-performance computing resources. Structural and energetic analyses would reveal how a protein is activated, how interactions change as a result of mutations, and what basis accounts for the drug efficacy. It can provide a rational explanation for success or failure of cancer treatment. The simulator is able to rank binding affinities of drugs to their targeted proteins, which is outputted as an array of drugs listed in order of their preferences. Perl scripts are used to construct the workflow in which a series of simulations is conducted. The scripts are executed from the front-end command line.

14 ObTiMA

- **Name of the tool/service** : ObTiMA – Ontology-based Trial Management Application
- **Institution(s)** : USAAR, FhG-IBMT
- **Contact person**: Holger Stenzhorn (holger.stenzhorn@uks.eu)
- **Short description**
  - **Type**: standalone, web application

- **Input data**: people enter manually data for a trial or (for retro-specitve data) they will be able to import them via CDISC
- **Output data**: export of collected clinical trial data via CDISC that can be used to store (long-time) the data tools and the type of interaction

- **Potential users**: bioinformaticians, researchers, clinicians, patients
- **Reference** : http://obtima.org

15 Ontology Aggregator

- Name of the tool/service : Ontology aggregator
- Institution(s) : IFOMIS
- Contact person : Ulf Schwarz (ulf.schwarz@ifomis.uni-saarland.de)
- Short description : The ontology aggregator will enale users to compile or build their individual ontological (or semantic) resources on the fly according to their specific needs out of existing resources available in the web (i.e. BIOPORTAL, OBO FOUNDRY) by extracting relevant parts combining and integrating them using p-medicine's health data ontology trunk (HDOT).
  - Type : probably web service or application
  - Input data : Input data are mainly semantic or ontological resources in the web, we will concentrate on those available in owl format. Interface to be designed for the specification of users semantic needs.
  - Output data : Ontology modules coded in owl.
  - Tools and the type of interaction : It could interact with the ontology annotation tool and the health data ontology trunk (HDOT).
  - Potential users : professional data managers

16 OpenStack Object Storage

- Name of the tool/service : "OpenStack Object Storage" cloud storage solution - extension by CDMI and Shibboleth support
- Institution(s) : PSNC
- Contact person (name, e-mail) : pukacki, dejw at man.poznan.pl
- Short description : CDMI extension for OpenStack Object Storage would be an implementation of the Cloud Data Management Interface (CDMI) specification - http://www.snia.org/cdmi Native DICOM files support - metadata will be extracted automatically and will be easily searchable through advanced search mechanism. National Data Storage integration - support for large, reliable polish nation wide distributed storage system which will play the role of the storage backend of the cloud solution provided for p-medicine project. Shibboleth integration to conform to the security standards chosen by the p-medicine project
  - Type : web service
  - Input data : HTTP REST type calls, input data or parameters would be a text or binary streams as well as structured inputs using JSON (lightweight text-based open standard)
  - Output data : Any binary or text files stored in the cloud storage as well as metedata stored as text documents using JSON notion
  - Interaction with other tools:
    - Data Warehouse through HTTP REST calls,
    - P-medicine portal through HTTP REST calls to provide users private data management
  - Potential users : IT experts, bioinformaticians, researchers, clinicians, patients, general public

- References :
  - OpenStack Object Storage - http://openstack.org/projects/storage/
  - CDMI specification - http://cdmi.sniacloud.com/
  - Shibboleth - http://shibboleth.internet2.edu/
  - NDS - http://nds.psnc.pl/

17 dcm4che DICOM server

- Name of the tool/service : "dcm4che DICOM server" - extension by OpenStack/CDMI and Shibboleth support
- Institution(s) : PSNC
- Contact person : pukacki, dejw at man.poznan.pl
- Short description : Shibboleth integration to conform to the security standards chosen by the p-medicine project. Integration with OpenStack infrastructure to store DICOM data and metadata .
  - Type : web service, web application
  - Input data : PACS server used as standard DICOM storage solution
  - Interaction : Data Warehouse
  - Reference : dcm4che - http://www.dcm4che.org/

18 User data management in cloud services

- Name of the tool/service : Portal web application for user data management in cloud services developed within p-medicine
- Institution(s): PSNC
- Contact person : pukacki, dejw at man.poznan.pl
- Short description : Web application will allow storing users data files directly in the project's cloud storage infrastructure.
- Type : web application
- Interaction : OpenStack Object Storage services , P-medicine portal

19 Oncosimulator service

- Name of the tool/service : Oncosimulator service
- Institution(s) : PSNC
- Contact person : pukacki at man.poznan.pl
- Short description : Web service wrapper for Oncosimulator application compliant with p-medicne workflow environment and with cloud computation infrastructure
- Type : web service

20 Optimized Oncosimulator application

- Name of the tool/service : Optimized Oncosimulator application
- Institution(s) : ICCS, PSNC
- Contact person : pukacki at man.poznan.pl
- Short description : Implementation of optimized Oncosimulator code that can be run in a parallel way and exploiting specific hardware architectures (eg. GPU)
- Type : standalone application
- Potential users : researchers

21 Oncosimulator environment

- Name of the tool/service : Oncosimulator environment
- Institution(s): PSNC
- Contact person : pukacki at man.poznan.pl
- Short description : Web application for running Oncosimulator application on dedicated cluster including some visualisation of results
- Type : web application
- Potential users : researchers

22 Workflows and pipelines for genomics data

- Contact person : Francesca Buffa (francesca.buffa@imm.ox.ac.uk)
- Institution(s) : UOXF
- Short description : Oxford is developing and will contribute workflows and pipelines for analysis of genomics data such as SNP arrays, miRNA arrays and sequencing data. We will also be contributing pipelines for integrated analysis of imaging data and genomics data.

23 Patient empowerment tool

- Name of the tool : Patient empowerment tool
- Institution that develops the tools : Philips
- Contact person : Anca.bucur@philips.com
- Short Description:
- Type: Web application
- Input data: Patient data, psycho-cognitive data, bio-bank data, clinical trials data, informed consent data
- Output data: Decision support, available clinical trials, bio-bank information, informed consent management
- Interaction with other tools: Clinical decision support tool – data exchange for decision support
- Potential users: Patients, clinicians

24 Data Mining Webapp

- Name of the tool/service : Data Mining Webapp
- Contact person : Michael Mock (michael.mock@iais.fraunhofer.de)
- Institution(s) : FHG-IAIS
- Short description : This service wrapps other data mining services, which could include R scripts, workflows or other services, and provides an interface which allows for using the data mining functionality from the portal, the decision support services or other clients
- Type : web application (REST based)
- Input data : flexible, e.g. id of the inner service/wf, some parameters and maybe small input data (relational)
- Output data : flexible id of a running service/workflow for querying the status and the results OR direct results (a predicted value) OR some plots OR some data stored in the data warehouse
- Tools and the type of interaction :
    - outside: service will be used by portal, decisions support or other clients

    - inside: service will interact with (wrap) other data mining services / workflows, scripts. these inner services will be mainly interfacing with the data warehouse.
- flexible, e.g., R, Galaxy workflow environment, literature mining services, data mining services
- Potential users : no graphical user interface for end users (just for demonstration purpose). it provides some kind of API to be used by clients. The internal services are partially based on GUIs to be used by the data mining service developers.

25 Genes-to-pathways correlation service

- Contact person : Anuj Sharma (a.sharma@biovista.com)
- Institution(s) : BIOVISTA
- Short description : The goal is to develop a RESTful web service which will allow users to correlate genes to pathways.
- Input : a list of gene names
- Output : pairs of genes and pathways.
- Each pair will be accompanied by a list of PubMed Ids where they co-occur in published literature. The service will utilize co-occurrence of genes and pathways in PubMed to generate the preliminary list. This list will then be refined using data from the KEGG pathway database, to generate the final list. The service will therefore provide a back end resource for any front end that may be used by a clinician, to quickly find pathways associated with a group of genes of interest and read literature associating them.

p-medicine initial scenarios

Scenarios, as described in D2.2 will be evaluated by using one or more forms.

1 Evaluation form

Scenario evaluation form (template)					
Name of evaluator(s)					
Evaluation start date			end date		
Task	Success Level[11]			Date	Note
	1	2	3	4	
Prerequisite tools and data	Description				Note
Scenario steps	Description				Note
Expected results	Description				Note
Rate of success and unconpleted task priority assignement					

[11] As listed in Figure 3 the four level of success are:

1. exceeds requirements
2. target range
3. minimally acceptable
4. unacceptable

Level done	1	2	3	4		
Comment on rating						
Uncompleted Task	assigned priority				due date	Note
	1	2	3	4		

2 Evaluation model

The complete evaluation model will be a document containing the following information:

Foreword and introduction

- Foreword
- Introduction

Scope

- Characteristics
- Level of evaluation
- Techniques
- Applicability

References

Terms and definitions

Inputs and metrics

- Input for the evaluation
- Data elements
- Metrics and measures

Interpretation of results

- Mapping of measures
- Reporting

Application procedure

- Definition of technical terms used
- Resources required
- Evaluation instructions

Documentation FOR QUALITY ASSURANCE REPORT (templates)

1 Software QA description

This document must be filled by every technical work package evaluation and validation delegate (local managers) and sent to WP15 for consolidating into a single report. Some of the sections have to be filled in only once, while others pertain to the evolution of the software during its development cycle and must be updated regularly. If it is natural to split the software developed in the context of a work package into several separate modules, multiple evaluation and validation documents can and should be filled.

Work package:	
Module:	
Author(s):	
Date:	

ABSTRACT:

KEYWORD LIST:

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0			
2.0			

Module identification

Purpose and scope	
Reporting period	
Definitions, acronyms, abbreviations	
Dependencies	
References	

Module requirements description

- Functional requirements

Functional requirement 1	
Introduction	
Inputs	
Processing	
Outputs	

Functional requirement n	
Introduction	
Inputs	
Processing	
Outputs	

- External interface requirements

User Interfaces	
Hardware interfaces	
Software interfaces	
Communication interfaces	

- Design constraints and Performance Requirements

(This section addresses constraints on the module, such as execution time, error propagation, memory requirements, etc...)

- Module attributes

(This section addresses issues related to the items in the six fundamental aspects not addressed elsewhere in the document. The subsection headers below are examples.)

  - Security
  - Portability
  - Maintainability
  - Other requirements

Quality assurance plan

- Software configuration and management tools used: (e.g. CVS, Bitkeeper, Bugzilla, make, ant,...), other specifics…
- Evaluation criteria for the 5 following ISO criteria: functionality, reliability, efficiency, maintainability, portability [The 6th , related to usability, is addressed in WP2]
- Software testing (for each module)
  - Verification (during development cycle)
    - Verification plan and design (describe scope, tasks and approach to component verification)
    - Verification log (verification actions and incident reporting)
    - Incident tracking (previous issues status (open/closed), new issues)
    - Verification summary (including an evaluation of the 5 criteria above: functionality, reliability,...)
  - Validation (when freezing a version of the module)
    - Validation plan and design
    - Validation scenarios
    - Validation log
    - Incident tracking (previous issues status (open/closed), new issues)
    - Validation summary. Include an evaluation of the 5 criteria: functionality, reliability,... (report also the fraction of requirements satisfied and those that are still missing).
- User documentation

(a “short-yet-complete” documentation including I/O specification, methods, error messages, ...)

BIBLIOGRAPHY

1. S. J. Zasada, A. N. Haidar, and P. V. Coveney, " On the Usability of Grid Middleware and Security Mechanisms.". Accepted for publication in the UK e-Science AHM 2010 theme issue of Philosophical Transactions of the Royal Society A, February 2011.
2. A. E. Abdallah and A. N. Haidar. "Usability Evaluation of Identity Management Schemes in Three Virtual Organisation Architectures". Int'l Journal of Information Assurance and Security, Special Issue on Information Assurance and Data Security, 4(6):560–570, July 2009.
3. A. N. Haidar, S. J. Zasada, P. V. Coveney, A. E. Abdallah, B. Beckles and M .A. S. Jones. "Audited Credential Delegation: A Usable Security Solution for the Virtual Physiological Human Toolkit". Accepted for publication in the VPH 2010 theme issue of Interface Focus of the Royal Society Interface Journal, March 2011.
4. ACGT FP6 project FP6-2005-IST-026996 Deliverable D13.1
5. VPH Toolkit, http://toolkit.vph-noe.eu/
6. ISO/IEC 14598, ISO/IEC 9126, ISO/IEC 25000, ISO/IEC 25041, ISO/IEC 25042, ISO/IEC 25043.
7. p-medicine – FP7-ICT-2009-6 - ICT-6-5.3-VPH - 270089, deliverables D2.1 and D2.2

Appendix 1 - Abbreviations and acronyms

SOA	Service Oriented Architecture
SQAS	Software Quality Assurance System
SCM	Software Configuration Management
QC	Quality Control
CAT	Custodix Anonymization Tool
DSL	Domain-Specific Language
CRF	Case Report Form/ Clinical Record Form
CRF	Case report form
DAkkS	National accreditation body of the Federal Republic of Germany
DICOM	Digital Imaging and Communications in Medicine
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers