User requirements for the evaluation of developed software and tools regarding usability criteria

Project Number: FP6-2005-IST-026996

Deliverable id: D2.3

Deliverable name:
User requirements for the evaluation of developed software and tools regarding usability criteria

Date: March 2008

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	ACGT
Project Full Name:	Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery
Document id:	D2.3
Document name:	User requirements for the evaluation of developed software and tools regarding usability criteria
Document type (PU, INT, RE)	INT
Version:	3.0 (pre-final)
Date:	March 2008
Authors: Organisation: Address:	Alexander Hoppe UdS Alexander.Hoppe@uks.eu

Document type PU = public, INT = internal, RE = restricted

ABSTRACT:

The present document describes the requirements for the evaluation of the usability of software/tools developed during the ACGT project and provides the criteria for the implementation of new software/tools in the ACGT platform. It is subdivided into four major parts. The first part (Section 2) provides a state of the art review of current standards both for the evaluation of usability criteria and for the evaluation of software by end users during the development process. The second part (Sections 3 and 4) describes the mechanism for the usability evaluation, emphasizing the user-friendliness of the software and the cost-effectiveness of the project. The third part (Sections 5, 6 and 7) defines the usability criteria and the use cases: the core tools and computer applications of ACGT are briefly reviewed, regrouped by their major target user group. In the fourth part (Section 8) the timeline for the evaluation in ACGT is defined. The QA procedures for the ACGT software components and the evaluation of the ACGT software can be found in Deliverable D13.1 (https://bscw.ercim.org/bscw/bscw.cgi/d334146/ACGT_D13.1_final.doc) While the scenarios already defined in D13.1 address individual components of the ACGT environment, the present document describes the evaluation of usability from a more integrated end-users perspective and provides strategies to guarantee the development of user-friendly software, tools and architecture.

The criteria and mechanism described in this document ensure a state of the art framework to integrate the end-users needs from the initial design period, through the development of demonstrators, up to the final end-users evaluation of the ACGT platform, considering the (sometimes very different) needs of the various end-user groups.

KEYWORD LIST: Usability, Evaluation, end-user, software and tools, use case,

context scenario, timeline.

MODIFICATION CONTROL			
Version	Date	Status	Author
0.1	19.11.2007	Int	Alexander Hoppe
0.3	08.01.2007	Int	Alexander Hoppe
1.0	28.01.2007	Int	Alexander Hoppe
1.2	22.02.2007	Int	Alexander Hoppe
3.0 (pre-final)	09.03.2007	Int	Alexander Hoppe
4.0			

List of Contributors

- Norbert Graf, UdS
- Thierry Sengstag, SIB
- Stefan Rüping, IAIS FhG
- Marie-Luise Christ-Neumann, IAIS FhG
- Dennis Wegener, IAIS FhG
- Radu Gramatovici, SIVECO
- Tomasz Piontek, PSNC
- Johan Karlsson, UMA
- Alexander Hoppe, UdS

Contents

Executive Summary

The present document provides an overview of the procedures, criteria and mechanisms used in the context of the project ACGT to evaluate the usability of the platform (software and tools) by end users.

In ACGT the developed software is evaluated following two major approaches:

- The first one described in D13.1 is the evaluation and verification of the ACGT platform by testing its components individually, by mean of specifically designed mini-scenarios.
- The second one is to assure that the software is working as expected from an end-user perspective (i.e. to ensure that it meets the demands of the target groups) by evaluating its usability as a whole, in a feedback loop between developers and users. This should guarantee the usability, and ultimately the actual use, of the software by the biomedical community.

This document addresses thus the evaluation mechanism of software as a whole, in parallel to the procedures for the components of the platform defined in D13.1.

Three different user groups active in the field of cancer are identified as major target groups in the present document: Clinicians and Healthcare professionals, Biomedical Researchers and Software developers, each having specific usability criteria in relation to the nature of their activities and associated tools. Specific sections of this document elicit those needs.

Based on existing standards in the field of software design (IEEE, DATech), this document defines reliable evaluation criteria and processes that can be used during and after the development of software in view of its optimization.

These evaluation criteria and processes are user driven from the beginning of the project to assure that all requirements of end users are covered during the process of software development

To assure the delivery of a high-class research environment, ACGT has to guarantee that only high quality software and tools are implemented in the platform. This document can hence be viewed as a set of guidelines for software developers, rendering explicit the criteria that have to be fulfilled by a “candidate” software (developed either inside or outside ACGT) to meet the standards of ACGT. By considering them in the development process, these evaluation criteria become an integral part of the quality-assurance mechanism in ACGT.

With the help of end users an assessment of the usability of the ACGT environment will be possible, even after publication. End users will be asked to evaluate the software and tools they use according to the usability criteria provided here to give a direct feedback to the developers, thereby ensuring the continuity of the optimization process.

Finally a timeline for the various evaluation activities is provided, to ensure a proper follow-up of the development process over the duration of the project ACGT.

1 Introduction

The use of computers, software applications and IT in daily medical life and in research is rapidly increasing. The main task to assure usability of the developed systems is to accomplish user needs. Generally, software is developed without evaluation during the development. To avoid this well known risk, it is of utmost importance to involve the end user from the design phase of new software, during the development process and after publication to secure an iterative evaluation of the software by end users.

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

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

Without taking the end user into account the risk of the software failing usability criteria is unpredictable high. At the worst end-users will not use the software or the tool leading to a serious loss of time, money and resources for the project. By developing usability criteria it is very important to know that most of the end-users are not used to evaluate software at all.

To assure the usability of software and tools, criteria have to be defined, that

- can be used as a guideline for end-users helping them to evaluate the software.
- provides an efficient feedback for the developer to optimize his software

To assure that the software used in ACGT will meet the high demands of the end users needs, the usability must be clearly defined and guarantee that:

- the software developed by ACGT is evaluated by the end users throughout the developmental period
- the software implemented in the ACGT platform fulfils the requirements for usability of the ACGT main target groups

Due to the high variety and complexity of the software to be produced/ integrated in the ACGT platform it is impossible to cover all possible aspects of the end-users evaluation for usability. The intention of this document is to give a general scope and basic templates for the evaluation process containing usability criteria. Individual specifications including the functionality of new software must be defined from case to case.

The usability criteria of software from the perspective of end-users are part of a top-down approach. Different categories and criteria are defined for end-users in terms of their suitability to achieve the intended goals with effectiveness, efficiency and satisfaction.

2 General aspects for End User Evaluation of developed software in ACGT regarding usability

2.1 State of the art

Standards related to usability can be categorised as primarily concerned with:

- the use of the product (effectiveness, efficiency and satisfaction in a particular context of use)
- the user interface and interaction
- the process used to develop the product
- the capability of an organisation to apply user centred design

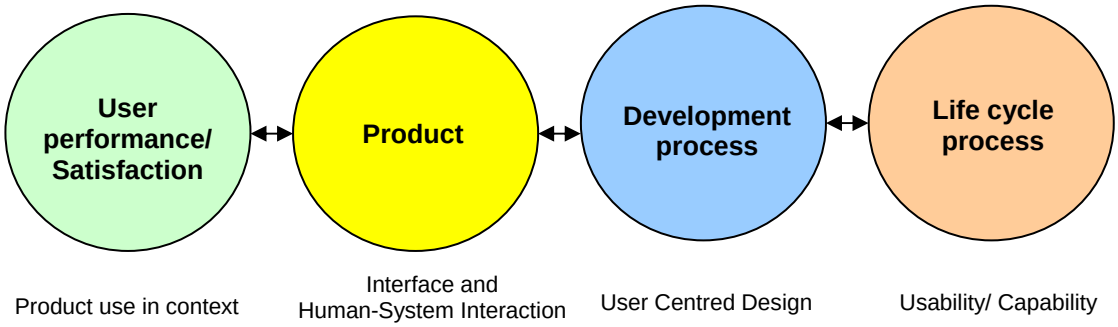

Fig.:1: Coherences in software development (1)

ISO Norm Principles and recommendations regarding usability

	Principles and recommendations	Specifications
Use in context	ISO/IEC 9126-1: Software Engineering - Product quality	ISO 20282: Usability of everyday products
	ISO/IEC TR 9126-4: Software Engineering - Product quality	
	ISO 9241-11: Guidance on Usability	

	Principles and recommendations	Specifications
Interface and interaction	ISO/IEC TR 9126-2: Software Engineering - Product quality - Part 2 External metrics	ISO 9241: Ergonomics of human-system interaction (since April 2006) Parts 3-9
	ISO/IEC TR 9126-3: Software Engineering - Product quality - Part 3 Internal metrics	ISO/IEC 10741-1: Dialogue interaction - Cursor control for text editing
	ISO 9241: Ergonomics of human-system interaction (since April 2006) Parts 11-17,110 Dialogue Principles	ISO/IEC 11581: Icon symbols and functions
	DIN 66285: Requirements for standard software	ISO 12119: Quality requirements and testing
Documentation	ISO/IEC 18019: Guidelines for the design and preparation of software user documentation	ISO/IEC 15910: Software user documentation process
Development process	ISO 13407: Human-centred design processes for interactive systems	ISO/IEC 14598: Information Technology - Evaluation of Software Products
	ISO TR 16982: Usability methods supporting human centred design	

Table 1: ISO Norm standards related to usability

Many standards are related to or affect the usability of computer software and applications. These standards have to be taken into account during the developmental process. For the evaluation of usability regarding developed software and tools the ISO 9241-11[(3)] guidance of usability is the most relevant standard today. It describes an objective, structured process to identify the users' requirements for the software and the mechanism to modify software applications and procedures with regard to the functionality and usability of the software. The process described in the ISO 9241-11 will be used as a guideline for the evaluation of the usability. To respect the complexity of the ACGT project, the standard IEEE 830[(4)] is used to comprise the needs of software developers.

2.2 DATech 2007

The „Trägergemeinschaft für Akkreditierung“ accredits certificate authorities in respect to the directives of the international organisations of standards (http://www.iso.org/)[2]. The DIN EN ISO 13407[3] is standard in Germany for „User-centred design of interactive systems” describing a prototyping software developmental process. The process consists of four main topics:

1. Use-Context: documented description of the relevant users and the daily work and work station
2. Specify requirements: the documented description of the use context is to align the needs of users to the software demands and the relevant software specifications
3. Describe solutions: this can be done in the form of prototyping and mock ups or other iterative processes
4. Evaluate the solutions: the prototypes are evaluated by expert-reviews or usability tests, online-evaluation or a mixture of them. Modifications for the next developmental step are based on the evaluation of the discovered variances and the lack of usability

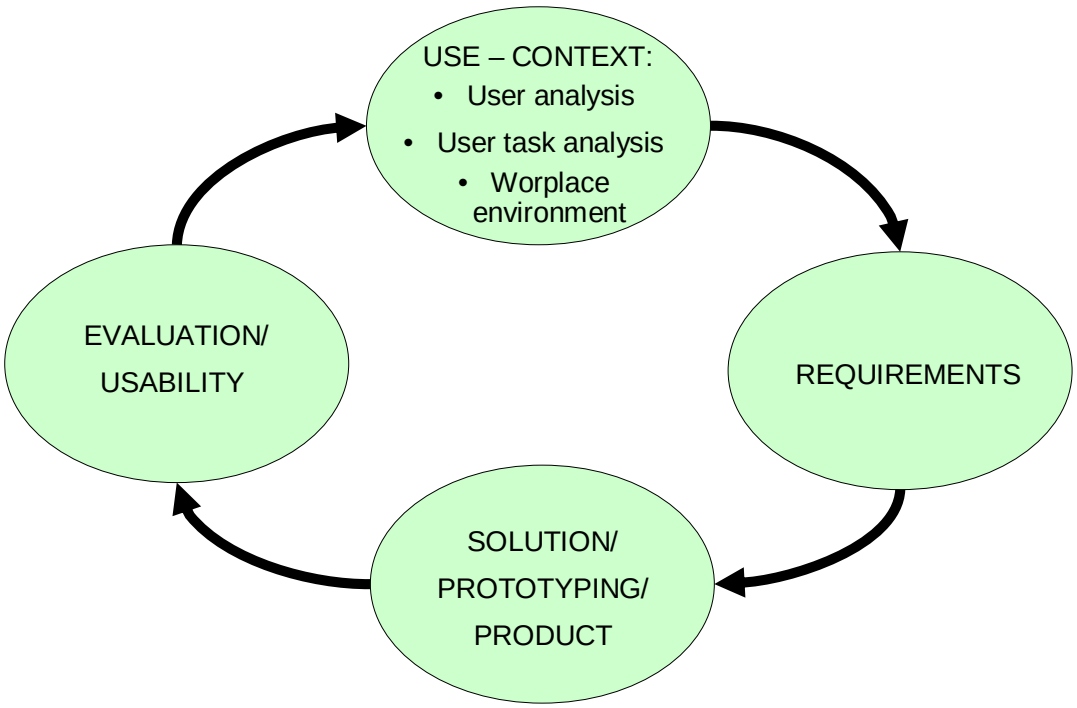

Fig.2: Schematic Structural process of ISO 13407

The international standard for usability of software systems ISO 9241 based on the EN ISO 13407 describes the interactions between humans and interactive systems. Part ISO 9241-11 (a pdf-file can be found at: http://www.datech.de/share/files/Leitfaden-Usability.pdf (no English version available) describes the requirements for usability and offers a guidance for practioneers to conduct usability tests of interactive software systems in particular their conformance with part 110 and part 11. Part 11 assigns three major directives:

- Effectiveness
- Efficiency
- Satisfaction

Part 110 is related to the user-interfaces of interactive computer systems. It assigns seven criteria to be fundamental for the dialogue for successful interactive computer and interface design:

- Suitability for the task
- Self-descriptiveness
- Controllability
- Conformity with user expectations
- Error tolerance
- Suitability for Individualisation
- Suitability for Learning

These criteria are essential for the usability of the tools and software developed in ACGT. An iterative evaluation process done by the major target groups to suit the user needs is recommended.

2.3 IEEE 830

The IEEE Standards[(5)] are developed by the Institute of Electrical and Electronics Engineers a society of engineers in the fields of electronics, software developers and IT with more than 360.000 members in 190 states. The standard 830 describes recommended approaches for the specification of software requirements. A short description can be found at: http://standards.ieee.org/cgi-bin/status?830-1998. It is based on a model in which the software requirements specification process is an unambiguous and complete specification document that should help:

- Software customers to accurately describe what they whish to obtain
- Software suppliers to exactly understand what the customer wants
- Individuals to accomplish the following goals
  - Develop a standard software requirement specification (SRS) for their own organisation
  - Define the format and requirements of their specific SRS.

- Develop additional local supporting items, such as an SRS quality checklist, or an SRS` s writers handbook

The main topics that have to be respected regarding the IEEE 830 are:

- Correctness
- Unambiguousness
- Completeness
- Consistency
- Ranking for importance and/ or stability
- Verifiability
- Modifiability
- Traceability
- Usability

ACGT will use these standards as basic quality criteria to test all prototypes. Usability criteria of DATech 2007 will be the standard regarding needs for end-user, while the needs for software specification can be done using the usability criteria of IEEE 830 norm.

2.4 Black box model

White or Glass box testing requires the knowledge about the program internals, while black box testing is based on the requirements of the end-users from the perspective of end-users. Because of the complex structure and architecture of ACGT both testing methods are useful and needed. Because of the high complexity of the ACGT software environment, it cannot be expected from end-users to understand it in detail. A white box approach is thus considered impractical. To secure the success of ACGT clinicians and researcher as end-users will use the ACGT platform only as a black box. There are several black box testing methods developed, which are useful for the testing of the ACGT structure. In respect to the user's requirements the black box testing and the white box testing is done by defining requirements in collaboration with all acknowledged end users.

2.5 White box model

Structural testing is an approach in which the internal control structure of a program is used to guide the selection of test data. It is an attempt to take the internal functional properties of a program into account during test data generation and to avoid the limitations of black box functional testing. Functional testing takes into account both functional requirements of a system and important functional properties that are part of its design or implementation and which are not described in the requirements. In functional testing, a program is considered to be a function and is thought of in terms of input values and corresponding output values [(6)].

2.6 Tasks for End Users evaluation in ACGT

2.6.1 The Complex structure of ACGT

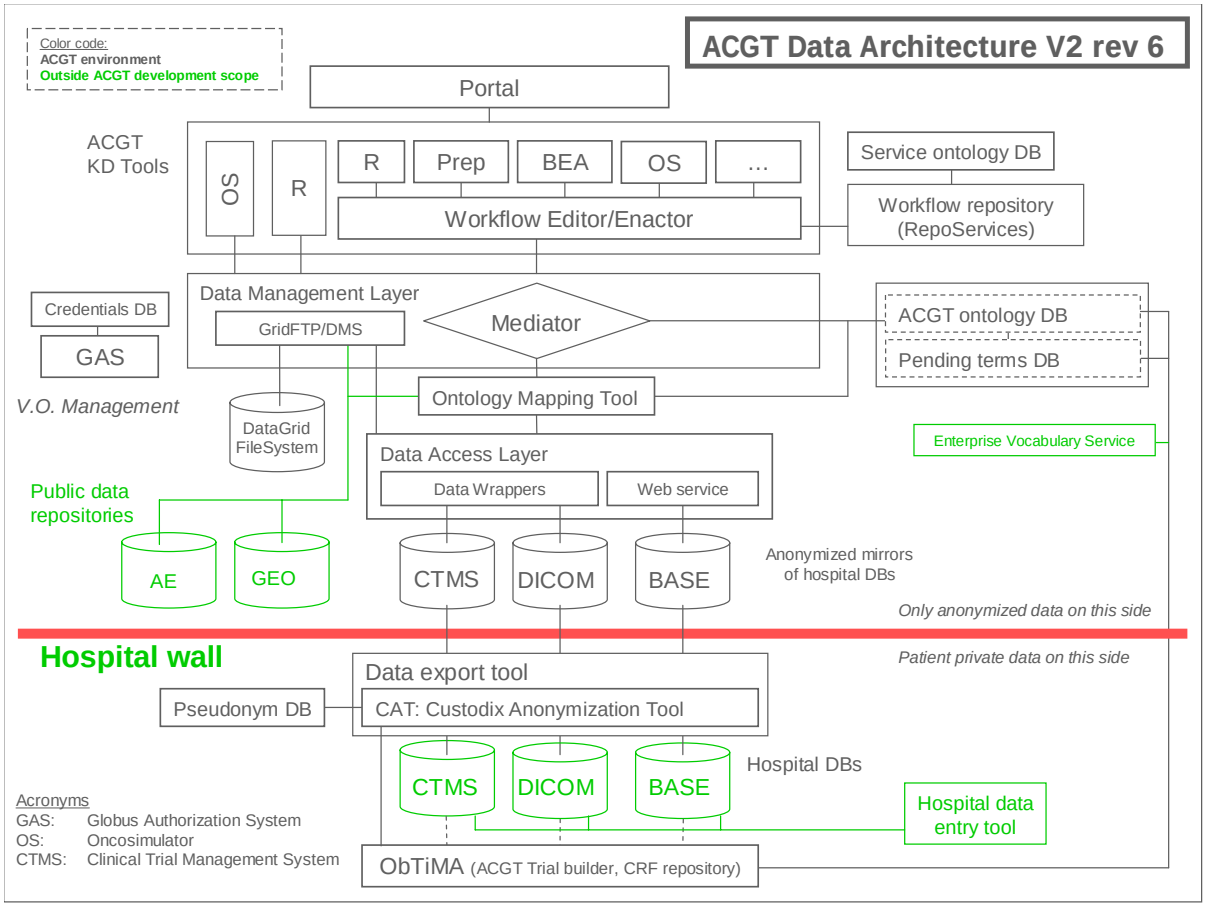

Fig. 3: ACGT Data Architecture V2 rev. 6, Thierry Sengstag, SIB, December 2007

The structure of ACGT has become very complex but well defined during the first period of the project. Because of the multiple tools and software implemented in the platform it is important to cover this structure from the end users view behind a self-explanatory interface. On the other hand the software developer must have a precise description what is needed to implement new software into the ACGT platform. To simplify the complex structure the tools were integrated in defined workflows and broken down into mini-scenarios (D 13.1.) For the first evaluation of scenarios, tools and software demonstrators for an end-to-end scenario will be described.

2.6.2 Software and tools as part of complex workflows

The aim of mini-scenarios (D 13.1) is to test functional units of the ACGT platform, such as accessing a database from the portal, track a patient history, anonymize patient data or integrate new tools in the ACGT environment. The software and tools are embedded in these mini-scenarios. To confirm the usability of the different tools

in the context of workflows during the development phase, short evaluation sheets provide a first impression of the usability and failures in the workflow.

Each evaluation form regarding the usability of the workflow should include:

- Workflow: (please describe the used tools/ software of the used workflow)
- Name of evaluated software/tool: (please use a template for each tool)
- Version Number: (Version number of the tool/ software)
- Name of the user:
- Function/ Profession of user: (please add: physician, researcher or software developer)

Task	Rating	Comments
Rate the accessibility level	O easy to use O hard O too complex	
Easy Log-in procedure	O yes O no	
Self-explanatory interfaces?	O yes O no	
Is the user-interface acceptable	O yes O no	
Is an on line help available	O yes O no	
If online help is available, is it sufficient?	O yes O no	
Are security mechanisms sufficient?	O yes O no	
Are the functionalities and next steps clear to you?	O yes O no	
If no, please precise which functionality was insufficient		
Is the software free of errors?	O yes O no	
Is the quality of outputs/results acceptable?	O yes O no	
Are all parameters required available by	O yes	

the single software/ the workflow?	O no	
Is the processing-time acceptable?	O yes O no	
Are information processed by the system acceptable?	O yes O no	
Have you encountered any problems with entering specific items?	O yes O no	
Are there formats you would like to use that are not supported?	O yes O no	
Was the workflow performed as expected?	O yes O no	
What problems in relation to the use of the workflow and your used IT/ computational software have you recognized?	O yes O no	
Is the workflow's result suitable for you?	O yes O no	
Access to information/ data was not possible (please specify)	O yes O no	

Table 2: Internal evaluation form for workflows and scenarios in ACGT

Individual evaluation sheets for the scenarios are already defined in D 13.1. The evaluation sheets in D 13.1 will be used to evaluate the functionality of the workflow, embedded tools and the platform during the development period.

2.7 Critical aspects

2.7.1 Actualization of End User Evaluation in ACGT regarding usability

The actualization of the end user evaluation regarding the usability will be part of D 13.2. "Intermediate Evaluation Report". The first evaluation results regarding the usability criteria and reporting forms described in this document will be part of the Deliverable D13.2. It will report on the actual usability status of the developed software and tools in ACGT.

2.7.2 Heterogeneity and Definition of user groups' needs/ Identification of main user groups

ACGT aims at several different target groups:

- Clinicians
- Biology/Bio-molecular Researchers
- Software developers and other vendors
- Regulatory Bodies
- Consultants
- Patients and Patient Advocacy Groups
- The General Public

To define the requirements for usability for developed software and tools, the following three major target groups have been identified which will use the software in the daily work.

2.7.2.1 Clinicians

Clinicians are one of the major target user groups for ACGT. To assure that the developed software is usable in the daily practice of physicians, their daily work and work place have to be defined and analysed. This will be done with the help of the usability engineer and reported and described in the context scenario. In the following the main specialisations of physicians are defined:

- Oncologists/ paediatric oncologists
- Surgeons
- Radiotherapists
- Pathologist
- Radiologists
- Reference physicians

Physicians play a twofold role as they can also be users in the user group of researchers.

2.7.2.2 Researcher

One of the main functionalities of the ACGT platform is to support biomolecular researchers in their daily work. ACGT should enable them to accelerate their knowledge discovery and understanding of bio-molecular processes. The amount of information and data in biomolecular research is increasing rapidly demanding intelligent software tools and platforms. Interdisciplinary and translational research has become important for this group because the biological information needs to be correlated with the clinical data of the patient. Today there is a wide range of web-based databases for genomic information but at the same time knowledge discovery has become more difficult because of a lack of adequate software and tools that can deal with this amount of data.

One of the functionalities of the ACGT infrastructure is to support biology and biomedicine researchers in knowledge discovery to accelerate understanding of biological processes and disease mechanisms.

2.7.2.3 Software developers

To advance the platform and the research tools and software for ACGT, it is one of the objectives to attract software developers to implement or design open source software for the ACGT platform. The recruitment of software developers and third parties can help to disseminate ACGT results. By doing so, the group of active users will increase the attractiveness and functionalities of the ACGT platform. In the Deliverable D 16.1 three distinct stakeholder groups were defined:

- Academic researchers who wish to research technologies and offer their own solutions
- Commercial Software developers who wish to offer their own products services or incorporate ACGT resources in their offerings
- Commercial medical or research instrument suppliers who wish to augment the capabilities of their offerings by making them compatible with selected ACGT modules/resources

For this target group the description of interoperability, codes and modules is one of the most important usability criteria, as they are a special target user group, who will not directly work with the developed tools but integrate own developments. The D16.1 can be found at:

https://bscw.ercim.org/bscw/bscw.cgi/d242867/D16.1%20ACGT%20Initial%20Exploitation%20Plan%20

2.7.2.4 Administrator

Needs to be defined

The objectives and the functionalities required from the main user groups vary for many concerns. While ACGT basic services, like mediator or portal are used by all end user groups, the individual processes of the end users are different. The main tools and software required by the different user groups are described in section 5, 6 and 7. A broad scheme of the architecture (detailed scheme Fig.3) and used functionalities by the main user groups is given below. The colours of the frames in the figure correspond to the different user groups. If a tool, software or functionality is used by more than one user group the frame is multicoloured.

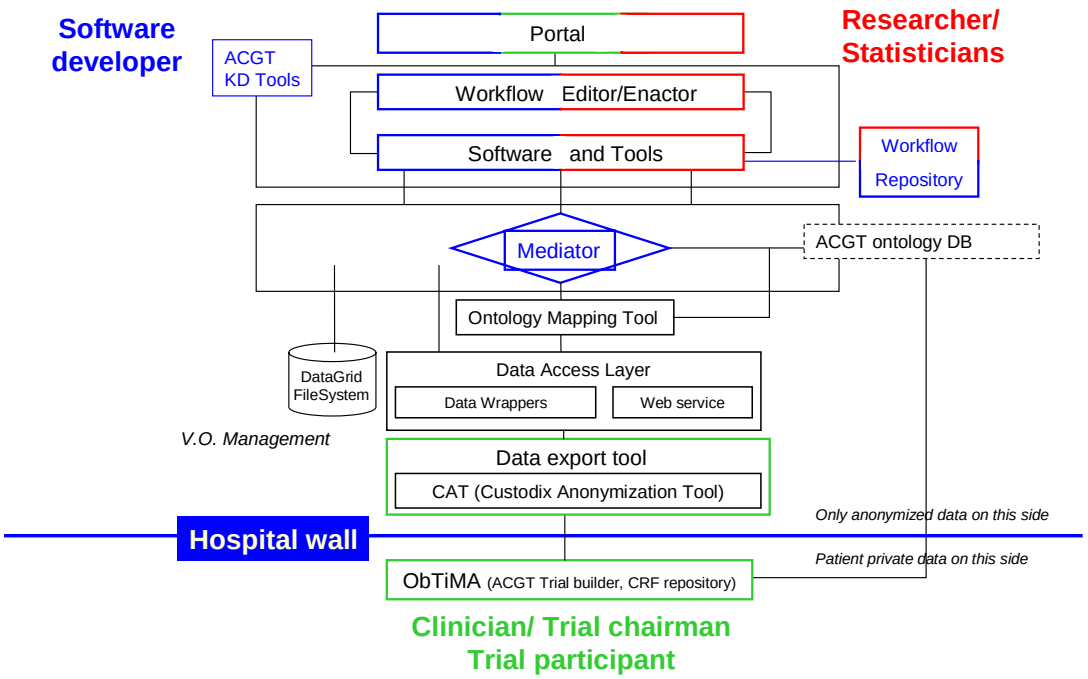

Figure 4: Simplified scheme of ACGT architecture and main user requirements

3 Requirements and Criteria for software evaluation in ACGT regarding usability

3.1 Software description

The software developed in ACGT can be separated in stand-alone tools and embedded software. For the usability of the software in ACGT the view of software-developers, clinicians and researchers has to be taken into account. As the clinician needs a functional and easy to use interface, the software developer has to get information on the interoperability and internals of the software and the tools. The software description should be regularly updated and written in a standardized format, to assure an easy use by conformity.

The standard for software description is the IEEE 1016 [(7)], which is even used to specify the software design of the developed tools. For a user-oriented description the standard is too complex and not useful for end-users looking for a brief description of the software. To assure conformity the software description should be structured using the following topics:

1. Introduction:
    1.1 Design overview
    1.2 Requirements Traceability matrix

2. Systems architectural design
    2.1 Chosen System architecture
    2.2 Discussion of alternative design
    2.3 System interface description

3. Detailed Description of Components
    3.1 Component 1
    3.2 Component 2
    3.3 Component 3

4: User interface design
    4.1 Description of the user interface
    4.2 Screen image
    4.3 Objects and Actions

5. Additional material

It is obvious that chapter 4 is most relevant to clinicians and biomolecular researchers and chapter 2 and 3 for software developers. Templates for software description can be found in Annex A

3.2 Usability criteria (ISO 9241-110)

The user's aim is to conduct his task supporting by the system in an efficient and satisfied way, e.g. to reach his results with minimal effort. Considering the ISO 9241-110 the DATech uses the dialogue principles to identify deficiencies of the software and describing which activities are difficult to perform by the user.

3.2.1 Suitability for the task

A software program or a tool is suitable when it supports the actions of the user and executes his task in an efficient and effective way that helps the user to ease his daily life and not complicate it. While most of the procedures done manually today ACGT will deliver a platform where automated workflows for analysis of user designed workflows will be done. Nevertheless this aim can only be reached when the developed tools, software and workflows process the data in a suitable processing speed and without complicated editing and sufficient support.

3.2.2 Self-descriptiveness

Self-descriptiveness is the degree to which a system or component contains enough information to explain its objectives and properties. The user should understand the tool by itself and the entries he should perform are obvious without external descriptions.

3.2.3 Controllability

Controllability plays a crucial role in many computational processes. In the broadest sense it is the ability to use the entire configuration of a system without severe errors or failures of the system. The user should never get the feeling, that the system fails or the computer crashes or a workflow is not performed adequately. The user knows in every situation what is expected from him as next step or how to return.

3.2.4 Conformity with user expectations

Per definition conformity with user expectations means acting according to certain accepted standards and the users knowledge and experience of software systems in means of consistency. In ACGT conformity is important for the usability on several levels:

- Workplace conformity: The user must have the feeling that the software of ACGT fits and supports him at his normal working place. For these needs the working places of the end users will be described in an objective way by the usability engineer in a report and aligned with the software developed for the special needs of the individual user
- Design conformity: e.g. the software interfaces of tools developed for ACGT should be designed in a similar way and recognised by the user as part of ACGT
- Format conformity: The formats used in the daily research and used for the ACGT platform are already defined. Nevertheless the need of implementation has to be iteratively evaluated.

3.2.5 Error tolerance

Input data errors often occur during the test phase and when a user learns about a new tool. While during the test phase the discovery of failures in the system is the aim, the usability of the software regarding the user will be strictly bound to the way he can interact with the system. A failure of user friendliness can be expected if:

- input data errors led to severe complication or handling problems,
- input data errors can not be cancelled,
- the user is not warned before an input data error causes a problem,
- no mechanism to avoid input data errors is available.

The user will judge the systems usability in the way data input errors are avoided. A loss of time or even worse of data will damage the reliability of the software. The user should have always the opportunity to solve problems in an easy and efficient way and get helpful instructions by error messaging.

3.2.6 Suitability for Individualization

Suitability for individualization of the software does not only mean to design the interface regarding colour and minor functions. The user should have the opportunity to set up settings that allow him to meet his individual needs. In case of the workflow editor for example, the user needs to store his workflows to use them again. Especially bio-molecular researcher often uses the same analysis redundant in different experimental tests. The more the needs for the specific user group are suited the more active users will use the ACGT platform.

3.2.7 Suitability for Learning

Suitability for learning describes a process where the user has the chance to learn about the software by simply try and error. The user should have the possibility to use the tools without damaging data, functions or the platform itself. There may be several equivalent one ways to perform an operation and at every time the possibility to cancel the progress without shutting down the whole program should be offered, e.g. by simply go back in the program step by step.

3.3. Mechanism and Evaluation strategy regarding usability

This section states the mechanism used to interview and evaluate the ACGT software from an end users perspective regarding usability criteria with the help of a usability engineer.

3.3.1 The development loop

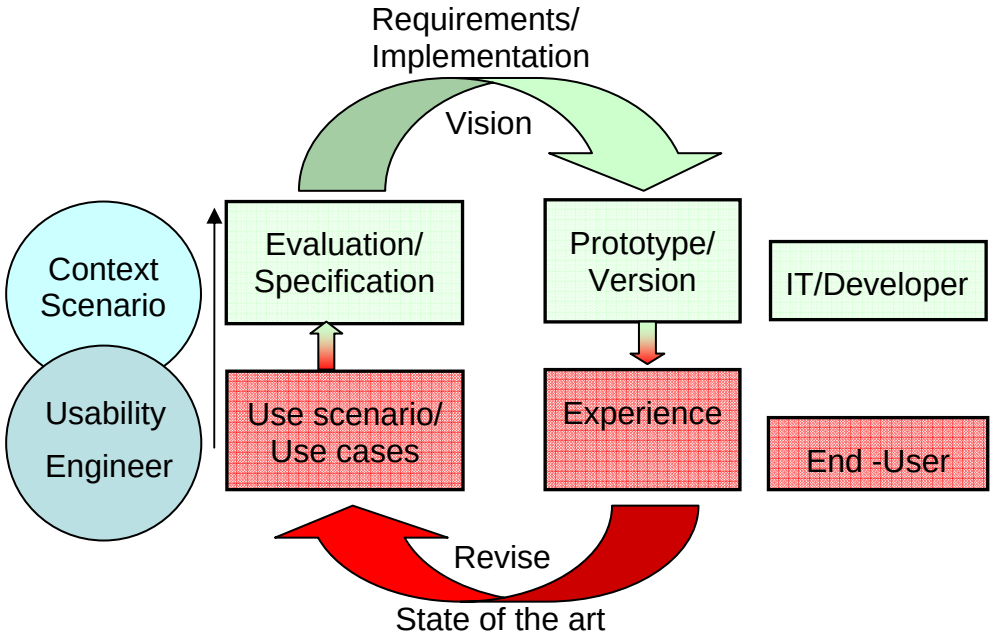

Figure 5: Development loop

During the first period of ACGT several scenarios were developed and modified, following the modular approach described in Deliverable D13.1. Beside a first integration of ACGT software components was also demonstrated with the “Farmer scenario” presented during the first formal review of the project. In order to focus more on a realistic use of the platform by end-users, so-called “end-to-end” scenarios have been retained. Considering the data and information available at the time of writing, these are:

- PseudoTOP scenario: This scenario is named after the ACGT TOP trial. As real trial data are not yet available at the present development stage of the project it was decided to simulate the trial with artificial (though realistic) data obtained from public databases or from simulation, while preserving the entire complexity of the data management system actually used in the TOP trial.

This scenario will involve microarray and clinical data mining using GridR scripts running on the ACGT environment. Complex database management is also a challenging aspect of this scenario, especially as regards data queries with the ACGT mediator which should look transparent to the end-user. The formal trial management aspects of this scenario will be limited though (limited application of ObTiMA).

- SIOP trial management scenario: This scenario has an emphasis on the data management side of the ACGT environment. It will in particular demonstrate the use of ObTiMA, the anonymization tool CAT, and the use of some analytical tools on specifically chosen data available in the context of the SIOP trial.
- MCMP scenario: This scenario aims at simulating a Multi-Centric Multi-Platform (i.e. multiple microarray platforms used in a trial), with an emphasis on the demand on large computational resources, but very limited consideration on trial management itself. In this scenario, microarray results stored on two databases will be exploited using algorithms running on a computational grid to investigate the medical evidence that can be obtained by considering the use of either homogeneous or heterogeneous microarray technologies in a clinical trial.
- MINDACT scenario: This scenario remains to be defined and its details will depend on the level of integration of EORTC as partner in the ACGT consortium.

The three first scenarios cover important and realistic aspects of clinical-trial data mining and all are expected to be implemented in the course of the third year of the ACGT project. These scenarios should thus provide a sound framework (availability of data realistically representing clinical trials) for assessing the integration and usability of a number of end-user tools plugged in the ACGT environment.

The details of these scenarios will be described in subsequent ACGT deliverables; in particular the PseudoTOP and SIOP scenarios will be detailed in D13.2 which will be available for the May 2008 review of the project.

3.3.2 Schematic Procedure of usability testing

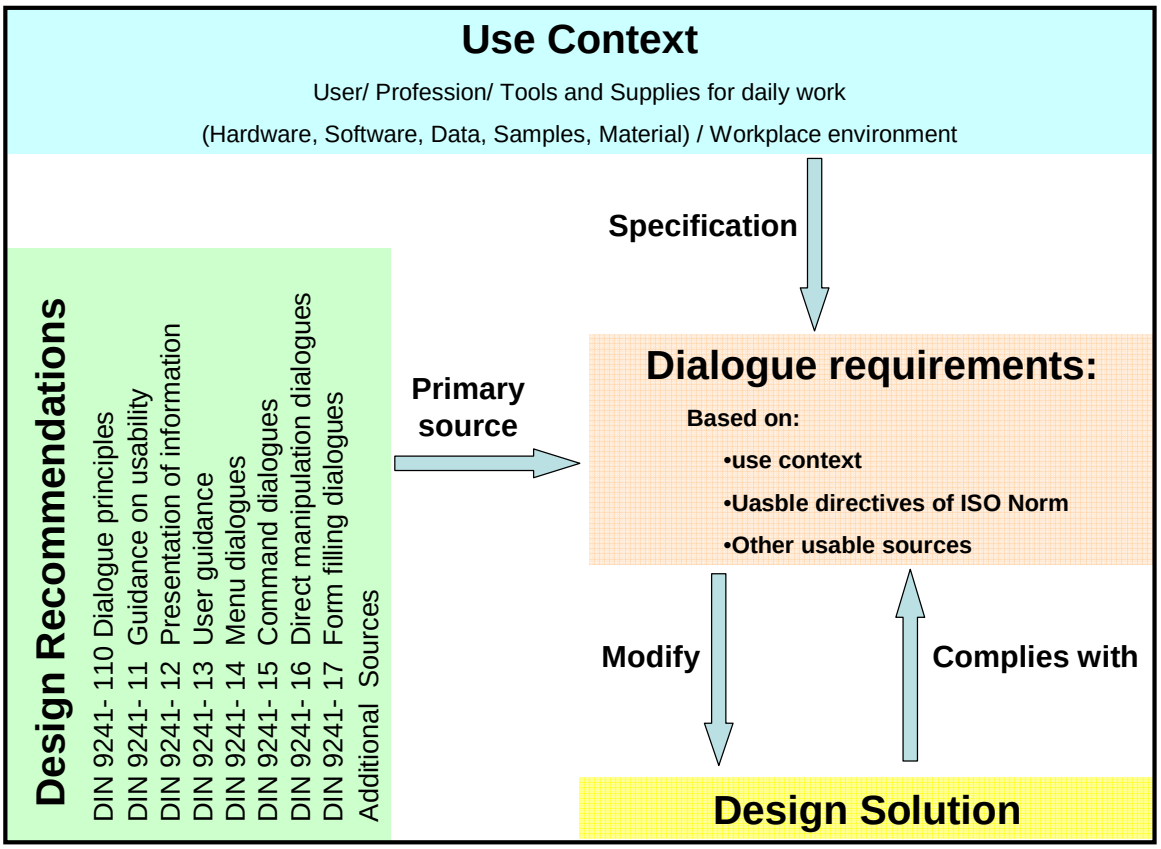

Figure 6: Framework execution regarding DIN EN ISO 9241-110

The framework execution (Fig.5) and the interviews will be done by a usability engineer in close collaboration with WP13, responsible for the evaluation and validation activities conducted in the project.

The major parts of usability evaluation for ACGT will be:

- **Context scenario:**
  - Understand the work of users and the organization of their working environment
- **Use scenario / Use cases:**
  - Use scenarios describe the action of the user with the system related to a predefined task and identify the user problems step by step.
  - A use case describes only the action of the user and the reaction of the system.
- **Validation and Evaluation of use scenarios/ use cases:**
  - Validation of defined use scenarios regarding usability

- **Validation and Evaluation of product:**
    - Validation of defined use cases regarding usability

3.3.2.1 Context scenario

The context scenario describes the task and the whole context of use in the real life situation of the employee/ end user taking into account:

Design Recommendations

DIN 9241-110 Dialogue principles

DIN 9241-11 Guidance on usability

DIN 9241-12 Presentation of information

DIN 9241-13 User guidance

DIN 9241-14 Menu dialogues

DIN 9241-15 Command dialogues

DIN 9241-16 Direct manipulation dialogues

DIN 9241-17 Form filling dialogues

The context scenarios are described by the usability engineer, who documents the interview in an objective report. This report is send to the end user for evaluation and after that to the software developer to achieve a common understanding of the whole task. The relevant key questions for the interview are shown in the templates in Annex D.

3.3.2.2 Use scenario:

In general a use scenario describes the user-system interaction with the aim to identify problems related to the interaction, to denote norm conformity and to discover critical incidences and weaknesses of the system.

The use scenario is based on the evaluation of the context scenario in which the minimal functions and requirements of the system were derived from the users implied needs.

During the use scenario the usability engineer is involved as a participatory observatory. The use scenario template is divided into four columns.

The first column describes the task to be executed by the user with the system. This task can be subdivided into several mini tasks to describe it in a more detailed way.

In the second column the

- action of the user
- “Thinking Aloud” (Users are asked to say whatever they are looking at, thinking, doing, and feeling, as they go about their task to get also an subjective behaviour and the expectation towards the system)
- and to identify critical incidence (behaviour of the user that results in an unsuspected reaction of the system or system failure)

are described.

The user questionnaire (see Appendix B) is protected by copyright by the DATech. To not breach the copyright ACGT is in contact with the DATech and has consent to use the questionnaire for the project after the request of FhG to Mr. Geis, member of the standardization board.

The third column reports on the reaction of the system in detail (errors, failure messages,..) and the fourth column analysis the single task process in respect to norm conformity or norm violation. An exemplary template for a reporting sheet is given below:

Task	**User action**	**System action**	**Problems/ Non Norm Conformity**
Part 1	Action	Reaction	e.g. Non Norm conformity
Part 2			
Part 2.1	Action Critical Incident	Reaction	e.g. Norm violation
Part 2.2	Action Critical incident	Reaction	No icon for planned task (e.g. no printing function) No task suitability

Table 3: Example of Use scenario reporting template

It must be stressed that only negative or failure behaviour in the execution of the task is described in the reporting forms. To avoid general problems the user should have a basic understanding in the use of computers, but he should not be familiar with the software. By doing so initial usage problems of the software can be detected.

Ideally the use scenario is performed in a professional usability laboratory using auditing software. The usability engineer records only the direct interaction of the user with the system, excluding the general behaviour except the thinking aloud. In ACGT this effort is hard to realise in respect to the several user groups. The solution will be a usability satellite session beside the regular ACGT meetings, where the prototypes can be tested by the end users of ACGT and the use scenarios are recorded by the usability engineer.

3.3.2.3 Use case

A use case is a use scenario without participatory observation and detailed documentation of the human-system interaction. Because of the high efforts of man power and costs of the use scenarios, it is obvious that clear defined use cases are powerful elements to evaluate the software, tools and the platform of ACGT.

The use case is defined based on the experience of the received design recommendations. A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Actors are parties outside the system that interact with the system [(8)]. A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied. It describes the sequence of interactions between actors and the system necessary to deliver the service that satisfies the goal. It also includes possible variants of this sequence, e.g., alternative sequences that may also satisfy the goal, as well as sequences that may lead to failure to complete the service because of exceptional behaviour, error handling, etc. The system is treated as a "black box", and the interactions with system, including system responses, are as perceived from outside the system [(9)].

A detailed description of the design of use cases and templates and references can be found at: http://www.bredemeyer.com/pdf_files/functreq.pdf [(9)]

For ACGT a Use Case for VO Management is planned:

This activity is very important. The project coordinator proposed a board. The members of the Use Case for VO Management can be found in Annex E

3.4 Usability Engineer (UE)

To formalize and objectify the usability criteria and advance and administer the usability performance as well as interactive processes in the project, the usability engineer will define the concrete usability concepts written in this document and has the functionality of an independent agent between the end users and the software developers. The engineer analysis the process during the whole development period from the first design defined in the user requirements D 2.1 (can be found at: D 2.1 https://bscw.ercim.org/bscw/bscw.cgi/d163285/ACGT_D2.1_FORTH_%20final.pdf and D 2.2 (can be found at: D 2.2 https://bscw.ercim.org/bscw/bscw.cgi/d328140/ACGT_D2.2_USAAR_final.pdf) For the success of ACGT it is of high importance that the software is self-explanatory and easy to use, because the main user groups have none or basic knowledge of computer systems or applications. The User Interfaces, as a gateway between project and end user, is of fundamental impact. The first step in the usability process, the user interviews, will start at the Consortium meeting in Milan.

3.5 Usability Engineer Process

Usability Engineering describes a pragmatic approach to interface design which emphasizes empirical methods and operational definitions of user requirements for tools concerning software ergonomics. To define these requirements the usability engineer defines the users needs in relation to his working place and the software concepts or prototypes developed. Extending as far as International Standards

Organization-approved definitions (see e.g., ISO 9241 part 11) usability is considered a context-dependent agreement of the effectiveness, efficiency and satisfaction with which specific users should be able to perform tasks [10]. In ACGT this process will be performed during the prototyping period to assure that the users' needs are satisfied. To objectify this approach, interviews with end users will be done by the usability engineer from FhG. After confirmation regarding content and correctness by the end-users the report is send to the software developer.

3.6 Online evaluation form

Usability reporting templates (see Annex C) are built to evaluate the usability and functionality of the developed tools and software and the whole platform in a standardised form. The use of standardised templates will allow an automatic analysis of the platform.

The evaluation of software and tools is an optional task to the end user when he uses the platform:

- The system should ask the user to evaluate the tool, when the user has performed a defined number of applications, tools and software. This has to be defined after the first trainings of the AVGT platform
- The system should ask the user to evaluate the portal when he logs out
- The evaluation form is freely accessible after the user logged in to the portal via a button in the task line

To attract the user to fill in the form, it must be as simple as possible. The amount of questions will be kept to a minimum that still allows a sufficient evaluation.

The header should provide a selection list, where

- the E-mail address, (via log in), date, the software name and version number is automatically filled in, if the end-user fills in the evaluation form after leaving a tool
- the name is automatically filled in and a selection list is provided with the available software of ACGT
- after the user clicks on the chosen software the actual version and the name of the software is filled in

A casket below the header gives the user the option to evaluate (in exceptions that can be done anonymous) and the entries in the header are deleted.

The target groups will receive different usability evaluation forms. The user group clinicians and researcher will evaluate the system regarding usability of the interfaces and the workflow performance while the software developer is interested in the interactions between his system and the architecture and standards of the ACGT platform. While the first user groups receive the evaluation forms through the portal, the software developer should receive it directly after implementation of the software.

4 Measurement and Quality control

4.1 Measurement and assessment

The measurement and assessment for the usability of the software and tools developed in ACGT should be categorised in ranking criteria that is based on a scale and free text for comments.

4.2 Ratings

Standardised ranking forms for the scenarios and developed software and tools are as described in the guidelines and the D 13.1.

4.3 Quality control and specifications

The quality control was first described in the D 13.1. The quality control is one major issue to secure the success of ACGT. There should be mechanism in accordance to the timeline and developmental loop that will assure that the developed software in ACGT is not only user-friendly but also done in a certain timeframe.

5 Requirements for the Evaluation of Usability of ACGT Core software for the user clinician

5.1 Introduction:

In the following three chapters a brief description of the actual software and tools, developed in ACGT so far, is given. The purpose of this description is dedicating the tools to their main user groups and defines the user groups for each tool. Because of the complexity of the ACGT architecture, this should provide a better overview of the single software and the target user groups. The evaluation of usability depends on the kind of user groups, as physicians and researcher evaluate the usability from the users' side, while the software developer understands usability in the interoperability of his product within the ACGT architecture.

Installation of ACGT software at participating hospitals

The installation of the ACGT software at peripheral hospitals is depending on structural, computational and personal capacity at the participating hospital. It must be possible for a researcher, physicians or other user groups, who have no computational background, to install, run, browse or use the software and computer applications on their computer. In D 13.1 the initial use scenario was described as "Software installation" (Chapter 3.1), mainly for the interoperability between software and portal.

From an end user perspective it is important that:

- There is an easy access via the portal
- It is a self-descriptive environment or sufficient online help is provided
- the user must know for each entry what is expected from him and what will happen next
- the programs can be run external without installation on the local computer, because the end-user does not always have administrative rights to install programs or run software and applications on the computer at his institution
- direct help functionalities and online help is available

Every tool developed in the ACGT should be ranked and evaluated by the end user regarding the usability criteria described in section 3. That means

- Software description
- The importance of self-explanatory software from the end users perspective
- Online help functionality (Yes/ No who is responsible for the update)
- The usability criteria in line with ISO 9241-11 as described in chapter 2 in this document
- Update possibilities

- Legal and ethical aspects
- Interoperability

5.2 Portal/ Virtual organization management interface

5.2.1 Product perspective

The ACGT portal is the interface of the ACGT environment, which attempts to provide an integrated easy-to use and up-to-date gateway to the ACGT tools and services.

5.2.2 Product functions

- Unique access point to the ACGT Grid
- Customizable client for ACGT services
- Usage of ACGT internal and external services
- Channel for ACGT dissemination and exploitation

5.2.3 User characteristics

- All user groups

5.2.4 Use case

- The portal as interface of the ACGT environment is part of most of the use cases

5.2.5 Basic requirements for usability already defined

Several basic requirements for the usability of the portal are already defined (Tab.3) in the table below. These basic requirements are essential for the usability of the portal. The further evaluation of usability will be done with the online evaluation forms that can be found in Annex A and the DATech evaluation forms that can be found in Annex C.

Requirements	fulfilled
Accessibility	
• Direct link from the website	
• Conformity with ACGT design (recognition)	
• User friendly account creation	
• Confirmation mail with next steps	
• Modify design	
• Easy way to receive login details and confirmation	
• Contractual agreements/ ethical agreement	
• Receive password (when lost)/ create own password	

• Flexible, user friendly GUI	
• Usable for every browser	
• Multi-language description	
• Online-help with step by step guidance and description of the use of the portal	
Data Entry	
• Monitoring of data entries	
• Password reminder	
• Comments can be given to each item	
• Understandable error messages (e.g. wrong password, unknown user instead of wrong password or user name) and password reminder	
• Clear and concise navigation and task lines	
• Self-explanatory browsing through the portal	
• Simple electronic signature	
Roles and Rights Management (Credential manager portlet)	
• User can use different roles using the same password and different credentials	
• Flexible, user friendly GUI	
• Multi-language description	
• Online-help (if applicable) with step by step guidance and description of the use of the portal	
File Browser Portlet	
• Online-help (if applicable) with step by step guidance and description of the use of the portal	

Table 4: Basic usability requirements for the portal

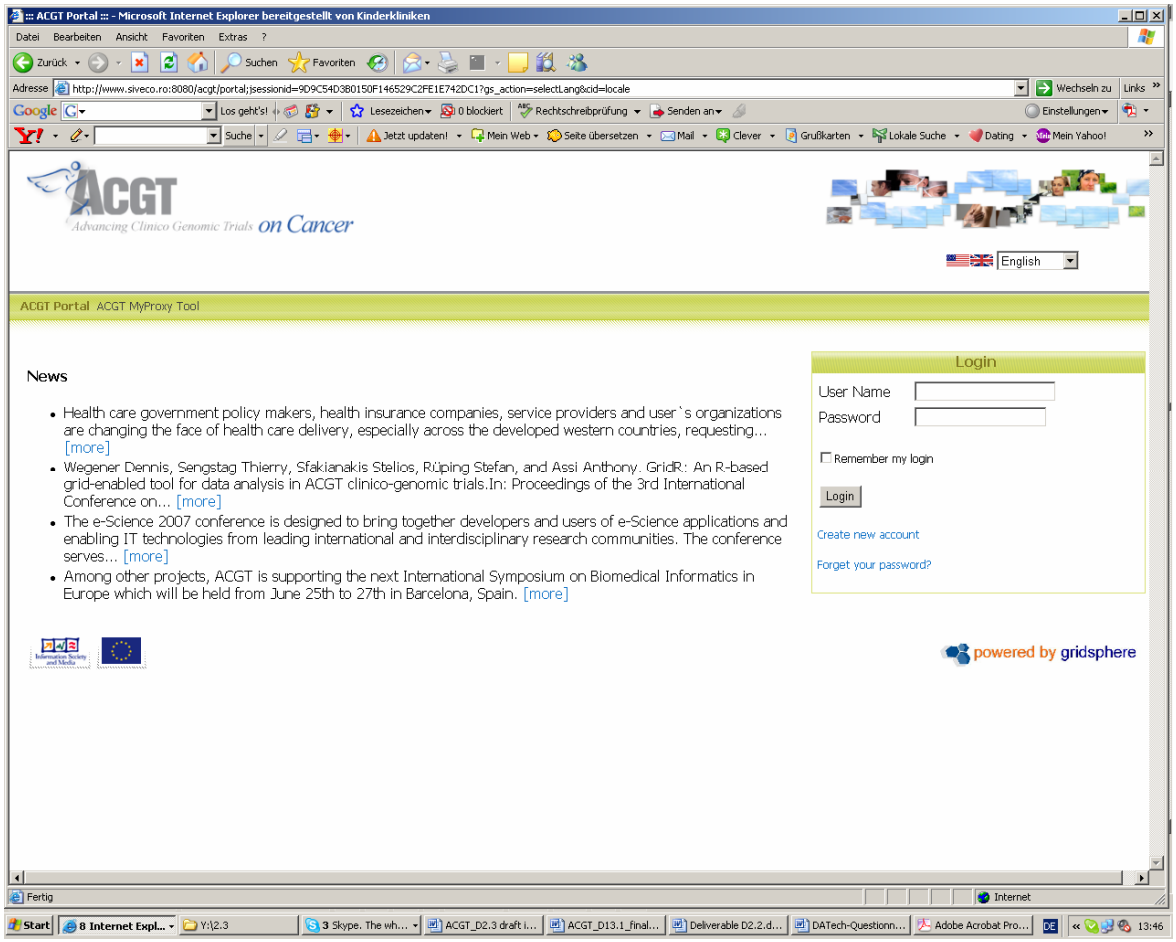

Figure 7: Screenshot of the ACGT portal log in (February 2008)

5.3 ObTiMA

5.3.1 Product perspective

Category: Stand-alone tool, Embedded Tool

The ontology based clinical trial management system for ACGT (ObTiMA) plays a crucial role within ACGT. ObTiMA will support the design phase of a clinical trial and will be a tool allowing an end-user to manage a patient within a clinical trial, to capture data, to report data and to query and analyze databases used in the trial in a standardized way.

5.3.2 Product functions

ObTiMA can be used in different levels:

- create and set up a new clinical trial according to the legal and ethical directives
- data entry via remote data entry (RDE) for managing patients in a clinical trial
- analysis of data in a single trial and across trials

5.3.3 User characteristics

- clinicians
- basic researchers
- patients (in a later version)

5.3.4 Use case

D13.1 Chapter 3.5.1: Use CRF editor to create ontology based CRF

D13.1 Chapter 3.5.1: Use trial builder to maintain the Master ontology

5.3.5 Basic requirements for usability already defined

Already defined in Deliverable 2.2, Page 153, Appendix 5

5.4 Text mining tool

5.4.1 Product perspective

Category: Stand alone tool, embedded tool

Biolab Experiment Assistant (BEA) is a literature-based environment integrating and cross-correlating a number of aspects of relevance to experiment design and allows clinicians and researchers supports to explore problem areas of their choice and discovering hidden links as well as in experimental strategies as discovering relations between biomolecular data, treatment, drugs and side-effects. Currently BEA extracts and correlates concepts in over 20 classes including: genes, pathways, diseases, cell lines, organisms, experimental procedures, reagents, medical tests and adverse events. That makes BEA and effective literature mining tool for clinicians and researchers. The functionality of BEA is described in detail in D 6.1, “Consolidated Requirement Analysis for Data Mining, Analysis and the Visualization Environment”, Chapter 4.6.,

5.4.2 Product functions

- literature search using genes, pathways, diseases, cell lines, organisms, experimental procedures, reagents, medical tests and adverse events
- user-friendly filter functionality, listing and display of literature search
- safe and recall, edit and export user sessions

5.4.3 User characteristics

- clinicians
- basic researchers

5.4.4 Use case

D 13.1, Chapter 3.4.9: Retrieve information about a set of genes using ACGT text mining tools

5.5 Oncosimulator

5.5.1 Product perspective

Stand alone tool, embedded tool

5.5.2 Product functions

After its prospective clinical refinement and validation, and in conjunction with a number of technological complementary components In Silico Oncology simulations should support the clinician in the selection of the optimal treatment of a patient taking imaging, histopathological, molecular and clinical data into account. The purpose is to simulate the response of the tumour cells to the administered treatment and compare it with reality. In a learning loop simulation and reality should stepwise overlap completely.

5.5.3 User characteristics

- Clinicians
- Researcher

5.5.4 Use case

D 13.1 Chapter 3.5.6 Use ACGT stored images, clinical data and molecular data as input for the Oncosimulator.

5.5.5 Basic requirements for usability already defined

The Oncosimulator is in a developing phase at the moment and used only for research and testing inside of ACGT. For the purpose of research only, there is little need to describe usability criteria at this point of time. Up to now there is no prototype available for end-users. As soon as the functionalities of the oncosimulator are defined the usability criteria can be defined.

5.6 CAT Custodix Anonymization Tool

5.6.1 Product perspective

Stand alone tool, Embedded tool

5.6.2 Product functions

Anonymization and Pseudonymization of all sorts of data, including Dicom data and data coming from microchip array analyzes in the lab.

5.6.3 User characteristics

- Clinicians
- Researchers

5.6.4 Use case

D 13.1, Chapter 3.1.4: Database anonymization (SQL/EHR and BASE)

D 13.1, Chapter 3.1.5: Database anonymization (SQL and BASE)

5.6.5 Basic requirements for usability

This will be modified and evaluated during the usability test period.

5.6.6 Evaluation of usability during the first development phase:

The CAT software developed by Custodix was primarily evaluated regarding usability in September 2007 by members of WP2 together with the developers at Custodix. To guarantee the usability it is important to formalize such meetings with the help of context scenarios, use cases and evaluation forms.

6 Requirements for the Evaluation of Usability of ACGT Core software for the user researcher

6.1 Work flow editor

6.1.1 Product perspective

Embedded tool

6.1.2 Product functions

The ACGT workflow editor is a key tool for the use of the ACGT tools and software for the user group researcher. The workflow editor follows a component based methodology consist of the Workflow Editor as an end user tool whose major functionality is the definition of new workflows and the Workflow Engine ("Enactor") as a network service that is responsible for the enactment of the stored workflows. The workflow editor should allow the end user to (*design*), manage and execute workflows with available tools and functionalities of the ACGT environment. Furthermore it provides maintenance of necessary metadata descriptions. As the basic requirements and functional usability requirements are already described in D 9.2, "Report on the implementation of the integrated ACGT environment and workflows" the usability by the end user is of highest importance as it is a major instrument for researchers using the ACGT platform.

6.6.3 User characteristics

- Researcher

6.6.4 Use scenario

D13.1 Chapter 3.3.14 Design a workflow, store it into a workflow db, retrieve it and execute it on test data

6.2 R based statistical analyses

6.2.1 Product perspective

Stand alone tool, Embedded tool

6.2.2 Product functions

R/Bioconductor (and its ACGT gridified implementation GridR) is a very general purpose script-based data analysis environment. It is primary goal is the statistical analysis of biomedical data, with a number of packages already dedicated to high-throughput technologies. Classical clinical methodologies, such as survival analysis or meta-analyses, are covered in standard packages.

Besides numerical data processing, R/Bioconductor also provides genomic data annotation (e.g. microarray and GO (Gene Ontology) annotation of genes) which allows the integration of simple biological knowledge with statistical results.

Another important feature of this software environment is the availability of a package which allows retrieving data directly from the GEO (Gene Expression Omnibus) repository of microarray results, thus facilitating meta-analysis.

In ACGT context an important category of users of R/Bioconductor are biomedical researchers who are developing algorithms for data mining which are tailored to the specificities of the clinical trials under scrutiny.

New analysis methodologies can easily be implemented through scripts and flexible plotting functions allow visualization of clinical data, although the interactive capabilities of the platform remain quite limited.

Once an algorithm has proven satisfactory it can be made available to the ACGT community, after wrapping in a workflow.

From a pure "usability" viewpoint, biomedical researchers developing new R analysis scripts require to easily edit them and to have the opportunity to repeatedly execute them on the same dataset. Usability criteria for R package developers are described in Section 7 of the present document.

6.2.3 User characteristics

- Researchers
- Clinicians (indirectly, through workflows)

6.2.4 Use scenario

A use case illustrating meta-analysis capabilities of the R/Bioconductor environment is proposed in Section 3.6 of D13.1

Application of R/Bioconductor in the context of the pseudoTOP trial will be shown during the demonstration of the First Integrated ACGT Demonstrator.

7 Evaluation of Usability Criteria of ACGT Core software for the user software developer

In this chapter, the evaluation of usability from the software developer's point of view is given, which means in detail the usability in the interoperability of his product within the ACGT architecture. For the target group software developer, the description of interoperability, codes and modules is one of the most important usability criteria, as they will not directly work with the developed tools but integrate own developments.

In the previous chapters, a brief description of the actual software and tools developed in ACGT so far was given. These tools and service may/may not or have to/ have not to be used for developing new software for the ACGT system.

As the ACGT system is designed as a high-class research environment, it has to be guaranteed that only high quality software and tools are implemented in the platform.

The ACGT system, based on a layered architecture as described in Deliverable 3.1," The ACGT Initial Architecture" can be extended by new components at different levels of the architecture. The most common scenarios for the integration of new software will address the following layers in the ACGT architecture:

- Services Layer: extension by ACGT compliant web services
- Resources Layer: extension by R packages, R functions/scripts

7.1 Implementation of new software - interfaces and documentation

7.1.1 General

From the user's point of view, the analysis tasks running in the ACGT environment might be quite complex. Because of that, users of the system have to be informed about the status of the different analysis steps and especially about runtime errors that might happen during the execution (e.g., because of system failures, failures in the program code of the tools and services, errors caused by using wrong file formats, etc.). Additionally, as the analysis tasks might be very long running, the system should provide a check pointing feature that allows for re-using intermediate results of a previous or aborted execution. Some of these needs are already addressed by the tools and interfaces provided by the ACGT system that are used when integrating new components. Software developers should keep in mind to also address these needs for allowing a seamless integration of components.

Additionally, the ACGT environment will provide the functionality of collecting provenance information during the execution of workflows and services. This provenance information will allow analyzing runtime behaviour, performance, fault tolerability and error handling, and can be used for improving software components at a later step.

7.1.2 Integrating Services (web services)

Web services in general are from the technical point of view self describing components, which means that it is directly obvious how the web services can be called, which methods they provide to the users etc. However, there are a lot of standards and technologies used in the context of web services. For the ACGT system, guidelines for ACGT compliant services were developed (see D 9.2., "Report on the implementation of the integrated ACGT environment and workflows") to ensure that new web services that are to be integrated can work together with the system. Moreover, a reference implementation of GSI-enabled web services is given (see D 4.1., "First production Grid layer") as example of an ACGT compliant service.

For making web services available in the ACGT environment, they have to be included into the ACGT metadata repository. The repository contains a tool registry where all details on the service, e.g. general metadata as the function/script name, the service's location, a textual description, information about the parameters, input and output data, the return type etc., are described. An API as well as a web portlet to register new services are in development and will be provided.

Services integrated into the ACGT system should provide an ACGT compliant logging mechanism and error handling that allow for checking the execution history as well as reacting to errors in a user-friendly way. Moreover, the creation of further meta data besides the return values of the services, e.g. provenance data, might be useful. The more the new component is interacting with the available functionality within ACGT (e.g., making use of extensive logging for error analysis, creation of provenance information about the execution, etc.), the more the users will profit.

Software developers are developing Java based web services in order to extend the ACGT environment should follow the common standards in modern object oriented software development, including e.g. test driven development (JUnits), following the Code Conventions for the Java Programming Language[1] and the Secure Coding Guidelines for the Java Programming Language[2], creating Java-Doc, etc. The java based components and interfaces of the ACGT system that are used for extending the system are also developed according to these guidelines.

7.1.3 Integrating R based components

The ACGT system enables users to access the R environment from the grid environment. New software components in the context of the statistical language R can be included either by developing R packages, that extend the R environment itself directly, or by developing R functions or scripts that can be included into the ACGT code repository.

[1] http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html

[2] http://java.sun.com/security/seccodeguide.html

Developing R packages:

For developing new software components that are deployed as R packages software developers should follow the official manual about "Writing R Extensions"[3]. Creating such R packages includes the generation of R documentation files. R objects are documented in files written in the "R documentation" (Rd) format, which is a simple markup language closely resembling LaTeX. This format can be processed into a variety of other formats for later usage, including LaTeX, HTML and plain text.

Each Rd file consists of the following three parts:

- Header - gives general information about the name of the file, the topics that are documented, a title, a short textual description and R usage information for the objects documented.
- Body - gives further information, e.g. on the function's arguments and return value.
- Footer - contains keyword information.

Header and footer are mandatory. There is also an official document for software developers specifying guidelines[4] for the creation of Rd files.

Developing R functions/scripts:

In ACGT, new algorithms for data analysis can be integrated as R scripts/functions. For making those scripts/functions available, they have to be included into the ACGT metadata repository. The repository contains a full and detailed description of the functions/scripts, including general meta data as the function/script name, a textual description, information about the parameters, input and output data and the return type as well as the actual R code.

An API as well as a web portlet to register new R functions/scripts is provided.

7.2 Bug reporting

Bug reporting is an important feature to be implemented in the ACGT platform to ensure continuous software improvement. The final platform to be used for that purpose was not yet selected, however it should follow the basic principles of usability in the ACGT environment, namely ease of use by the end-user, traceability of bug reports and user feedback.

[3] http://cran.r-project.org/doc/manuals/R-exts.html

[4] http://developer.r-project.org/Rds.html

8 Timeline for Usability testing in ACGT

8.1 Interviews and context scenarios

The interviews with the different end user groups started at 05-07.02.2008 at the consortium meeting in Milan. Marie-Luise Christ-Neumann from the FhG, St. Augustin, Germany, carried out the interviews besides the meeting. Members of the three main user groups participated.

8.2 Internal testing and evaluation of the ACGT tools and software

The internal testing and evaluation of the ACGT tools is an iterative process, driven by the usability engineer. The usability engineer will help to modify and optimize the tools tested in the context of use scenarios on the basis of the scenario evaluation results.

8.3 Evaluation of usability of the demonstrators

The evaluation forms for usability can be used during the test period and the preliminary review of the demonstrator. When the end-to-end user scenario is defined, the review forms will be adapted to match the new reality of the testing framework. After the first integrated demonstrator will have been made available, a workshop organized jointly with WP13 and aiming at the assessment of the ACGT platform “as is” will be held.

8.4 Evaluation of ACGT tools and Infrastructure by the end users community

The evaluation of the ACGT environment by the “external” end-user community will take place after the final architecture of the system will have been decided. It is likely that this will take place during the last reporting period of the project.

Appendix.A Online Reporting forms for end-user evaluation regarding usability criteria (general)

Header of online reporting template:

Description	
Software	
Version	
Author of Evaluation (E-mail Address)	
Profession	

Usability (user group clinicians and researcher)

The following report form is the short version for an evaluation to be filled in by users as described in chapter 3. The possibility to provide the long and more specific evaluations form described in Annex C is also possible.

Question	Answer	Comment
Was it easy to access the tool?	O yes O no O not known	
Is the software well documented?	O yes O no O not known	
Is the GUI sufficient?	O yes O no O not known	
Is a user manual provided?	O yes O no O not known	
Is the software well documented?	O yes O no O not known	
Is the user interface self-descriptive/	O yes	

Self-documenting?	O no O not known	
Are meaningful error messages provided?	O yes O no O not known	
Is the user interface intuitive?	O yes O no O not known	
How much teaching is necessary for using the software (hours) without problems		
Is it easy to perform simple operations?	O yes O no O not known	
Is it feasible to perform difficult operations?	O yes O no O not known	
Do widgets behave as expected?	O yes O no O not known	
Is the user interface responsive and fast enough?	O yes O no O not known	
Are relevant data formats supported?	O yes O no O not known	
Is the design of the user interface sufficient?	O yes O no O not known	

Usability questionnaire for user group software developer after implementation of the software

The questionnaire for the user group software developers aims to the software performance and interoperability of the implemented software and the ACGT environment.

Question	Answer	Comment
Was it easy to integrate a new software component into the ACGT system?	O yes O no O not known	
Was there an adequate documentation or user guide provided on how to integrate the new software component into the ACGT system?	O yes O no O not known	
Is the design cohesive, i.e., each module has recognisable functionality?	O yes O no O not known	
Are the security mechanisms provided by the ACGT systems adequate?	O yes O no O not known	
Does the software interact with the credentials of the ACGT environment without problems?	O yes O no O not known	
Is it possible to enforce security policies?	O yes O no O not known	
Can the software withstand attacks that must be expected in its intended environment?	O yes O no O not known	
Is the software free of errors that would make it possible to circumvent its security mechanisms?	O yes O no O not known	
Does the architecture limit the impact of yet unknown errors?	O yes O no O not known	
Have the tools and services of the ACGT system that are used by the new component been optimized for speed?	O yes O no O not known	

Is input data checked for the correct format?	O yes O no O not known	
Is exception handling provided?	O yes O no O not known	
Does the system allow for a change in data structures?	O yes O no O not known	
Using an API for integration, are variable names descriptive of the physical or functional property represented?	O yes O no O not known	
Using an API for integration, do uniquely recognisable functions contain adequate comments so that their purpose is clear?	O yes O no O not known	

Templates for Software design description

(http://www.erichsteiger.com/downloads/sdd_en.dot)

[Your company name]

[Your products name]

DD/MM/YYYY

Product:	[Your products name]		
Filename:	C:\sdd_en.doc		
Audience:	[your audience for this document]		
Date	**Version**	**Author**	**Description of Change**
[date of change]	[new version]	[your name]	[describe what you have changed in this document right now]

I. Index

Appendix.B Interview Forms

ErgoNorm

User Questionnaire "Work & Software"

Dear user

With this questionnaire we want to learn more about your personal opinion on the work with your computer used in your daily work. It is only you who can estimate how well the computer supports you in your work situation. It is a matter of finding out which activities are difficult to perform with the software in question, which are the steps that annoy you or leave you puzzled.

Maybe you are no longer aware of the deficiencies of the software during execution of your work because you have become accustomed to them, or maybe you think, "That's just the way it is." The questionnaire helps you to identify and name those weaknesses in the software. Your answers to the questions help to capture deficiencies in quality. The aim is to improve the computer to suit your needs, and therefore ease your work at your workstation.

All data will be collected anonymously, so that none of your statements can be traced back to you personally.

Handling of the questionnaire

Probably you use the computer to execute different and self-contained tasks. Please be sure to fill in the questionnaire with respect to the execution of the following task:

_____

_____

_____

Before you start to fill in the questionnaire please read through all the questions first. You will notice that all questions point to very useful features the computer should have. When completing the questionnaire, it is important for you to think about the task initially described. You should only answer those questions which are important to this task. If you think that a question is not concerned with that task, mark the answer "question does not apply". When filling in you can also indicate deficiencies. If you think them to be very disturbing, please mark the corresponding item.

Please don't start filling in the questionnaire until you have carefully read all the questions. It has proved valuable to complete the questionnaire continuously over a period of time. If you come across problems during your work, insert them in the appropriate position.

Please send the completed questionnaire (on paper or online) to the following address:

Fraunhofer IAIS
Marie-Luise Christ-Neumann
Schloss Birlinghoven
D - 53754 Sankt Augustin
Marie-Luise.Christ-Neumann@iais.fraunhofer.de

Your comments and suggestions are also welcome, even if you have already returned the questionnaire.

Description of task

(Please remember your special task when filling in the questionnaire.)

Suitability for the task

A program is suitable for a task when it is usable for the completion of your special kind of activity. "Usable" means, that all activities you have to perform are supported by the system in an effective and efficient way. The program should be a helpful tool not disturbing you by making your work harder or more complicated in some situations.

1.)

Has the program all the features required for your task?	O Yes O No O Question does not apply

If no:

Please indicate the dialogue step which makes you wish, that the program "can do more" than is possible now.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

2.)

Do you have to do redundant input actions or dialogue steps?	O Yes O No O Question does not apply

if "yes":

Please indicate redundant input actions and dialogue steps

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

3.)

Is it possible to facilitate repeated entering of data or text?	O Yes O No O Question does not apply

if "no":

In which situation do you wish that you do not have to enter the same thing again and again?

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

4.)

Do you think that the effort required to achieve the results of your work is appropriate?	O Yes O No O Question does not apply

if "no":

In which situation have you ever thought "This could be achieved with less effort"?

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

5.)

Do you think that you have to do task steps which should be done by the program?	O Yes O No O Question does not apply

if "yes":

Please specify these tasks.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

6.)

Do you have to enter values and text that the computer could really know?	O Yes O No O Question does not apply

if "yes":

Please describe the situations which make you think e.g.: "The computer should really know by now. Why must I write this once again?"

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

7.)

Do you have to go some other way or use tricks to achieve your working results as intended?	O Yes O No O Question does not apply

if "yes":

Please describe the situations where you play tricks on the system in order to achieve the intended result.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

8.)

Do you get help by the program which actually helps you?	O Yes O No O Question does not apply

if "no":

Specify the situations where the help information has not helped you.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

9.)

Does the program fit to your forms and current formats?	O Yes O No O Question does not apply

if "no":

Describe the activity where the program does not fit to your paper forms or formats.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

Self-descriptiveness

A program is self-descriptive when you are always informed of what the computer is doing and what is expected as your next input or reaction. This means, among other things, that all of the feedback is comprehensible and that you always know where to enter the next input and you always understand what consequences follows from your input.

10.)

Is the information you need to perform your task structured clearly on the display?	O Yes O No O Question does not apply

if "no":

Please specify the information you need but which is not available on the display "at a glance".

._____

O I feel this is very disturbing

11.)

During your work with the program can you recognize which input is expected from you next?	O Yes O No O Question does not apply

if "no":

Please briefly describe the situation where you are not sure about the next step to do with the program.

O I feel this is very disturbing

12.)

Are system messages always comprehensible to you?	O Yes O No O Question does not apply

if "no":

Identify the situations where you noticed messages you do not understand.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

13.)

Are you warned before you perform actions that cannot be undone by the software?	O Yes O No O Question does not apply

if "no":

Please specify the situations where you were not warned by the system.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

14.)

Does the help function really help you when a dialogue step or menu item is not entirely clear to you?	O Yes O No O Question does not apply

if "no":

Describe the situations where you do not understand the help text._____

O I feel this is very disturbing

15.)

Do you often have to ask colleagues or look up in the manual to continue with your work?	O Yes O No O Question does not apply

if "yes":

Please describe situations where you could not continue without the help of colleagues or a manual.

O I feel this is very disturbing

Controllability

A program is controllable if you can freely determine the sequence of your work steps. If it is required in your work situation, you can interrupt your work with the computer and resume work again without loss of previously attained results.

16.)

Can you execute your work steps in the order which makes most sense to you	O Yes O No O Question does not apply

if "no":

Please describe the work steps where a different order seems to make more sense.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

17.)

Is there sometimes a (re)action of the program you do not want at that moment?	O Yes O No O Question does not apply

if "yes":

Please specify the behaviour of the program which occurs unintentionally.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

18.)

Can you interrupt a task on demand and resume later, without having to re-enter everything?	O Yes O No O Question does not apply

if "no":

Please explain, in which situation you lost data already entered by a break?

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

19.)

Can you undo a working step when appropriate for your task performance?	O Yes O No O Question does not apply

if "no":

Please identify the situations where undoing a dialogue step would be advisable.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

20.)

Do you feel slowed down by the program in the pace of interaction, e.g. due to long response time?	O Yes O No O Question does not apply

if "yes":

Please describe the situations where you would like to work faster.

_____
_____
_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

Conformity with User Expectations

A program conforms with user expectations when you are not "surprised" by the program. Such a surprise can be, for example, a function being in a totally different position in the menu as you would have expected it, or tasks which cannot be performed as usual.

21.)

Do you find menu items or functions where you think they should be?	O Yes O No O Question does not apply

if "no":

Please specify the specific location in the menu or in another matrix, where the arrangement of information does **not** meet your expectation.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

22.)

Are you still sure during waiting periods that the program continues to work?	O Yes O No O Question does not apply

if "no":

Please specify the situations where you are not sure if the program is still working, for example, when the program needs a very long time to store data.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

23.)

Are you sometimes surprised at how the program reacts to your input?	O Yes O No O Question does not apply

if "yes":

Describe the situations where you are surprised about the reaction of the system.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

Fault tolerance

A program is error-tolerant when the intended result can be achieved despite evident errors in input with either no or minimal corrections by the user. This means, it has to be allowed to mistype or to make a wrong working step without causing a system crash or having to correct the mistake with great effort. In addition, you should be noticed by the program when an error occurs and hints for possible correction should be given to you.

24.)

Do you get correction hints on an incorrect input?	O Yes O No O Question does not apply

if "no":

Please specify situations where you might wish that the program proposes a correct input?

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

25.)

Can you recover from an incorrect input with minimal effort?	O Yes O No O Question does not apply

if "no":

Please describe briefly the situations where the effort for recovery from of an erroneous input appears to be too high.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

26.)

Does the program always work robustly and reliably during the execution of your task?	O Yes O No O Question does not apply

if "yes":

Please describe briefly the situation in which the effort for the correction of incorrect input is not affordable?

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

Suitability for Individualization

A program is capable of individualization when you are able to adapt the interface software for your individual needs.

27.)

Can you customize the computer so that you can read and work more comfortably?	O Yes O No O Question does not apply

if "no":

Indicate the places where working with the program is difficult for you.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

Suitability for Learning

A program is suitable for learning when it allows you to explore the program without having to be afraid of spoiling something. Additionally, you should get relevant information by the system which you need in your opinion to understand the program better.

28.)

Does the program allow you to learn by "trial and error"?	O Yes O No O Question does not apply

if "no":

Please describe the "punishment" which you already got when exploring the program.

_____
_____
_____
_____
_____
_____

O I feel this is very disturbing

The last part of the questionnaire is reserved for your personal observations. There is room for further criticism of the program or for the problems you could not write down in special parts of the questionnaire.

Thank you for your effort and help to improve the ACGT platform!

Appendix.C Key Questions for describing and structuring user performance in context

Introduction	1. Describe your daily work in one or two sentences. 2. Which tasks compose your work with the computer (list typical key tasks, which are time-consuming or frequently occurring or very important)? Which of the key tasks should be supported by the software? 3. How work is organised (e.g. in various tasks, as a sequence of tasks, as repetitive single task)?
Assumptions (or pre-condition)	4. What kind of qualification is needed for performing the tasks (for task completion / for using software)? What kind of skills are missing? 5. Who or which event decides what to do? (Who selects your jobs? jobs are performed autonomously, work is divided, data is needed from colleagues or external sources) 6. Which media or devices are necessary (for task completion / for software use)? Which of them are missing, which are desired additionally?
Routine activities (or usual performance)	7. Which working steps are executed? 8. Which working steps are performed repeatedly? (Automated execution desired / necessary)? 9. Which working steps are executed by the software? Can the user control the autonomous process / is control allowed / desired / required? 10. Are several users working in parallel with/ on the same object (e.g.

	transaction, file, document, data record)? 11. Is there a defined sequence of working steps? If so, how is it composed? (More flexibility needed / desired?) 12. Which are the results / partial results and how are they used / continued? 13. Which kind of feedback does the interviewee get concerning his working results and effects?
Special features during work performance	14. Which kind of interruptions appear? Why do they appear? When do they appear? (Organisational / Social / Technical)? 15. How are mistakes reported back and solved (Organisational / Social / Technical)? 16. Which important special cases have to be considered (Respectively comes up in the user's mind spontaneously; e.g. division of work / collaboration)?
Organisational conditions	17. Which organisational aims are defined for the working tasks? 18. Are there mechanisms to control the efficiency of work? (If yes, which ones? Are they necessary?) 19. Which overview has the user with respect to the overall workflow? 20. Which changes are expected or desired by the user considering the performance of work? Are there any suggestions from the interviewee? Visions! 21. Which results / working steps affect third parties (e.g. customers) directly? And which are the consequences? 22. Which are the stress factors and how are they handled?

Other comments to critical incidents which already occurred	Put examples in here, when the interviewee tells something about critical incidents concerning the software during the interview. Usually such problems analysed within use scenarios.

To derive requirements and test cases. Only this derived material is provided to the project members.

Appendix.D Board Members for the Use Case for VO Management

Name	Institution	City	Country
Stefan Castille (Chair)	Custodix	Ghent	Belgium
Norbert Graf	UdS	Homburg	Germany
Francesca Buffa	UOXF	Oxford	UK
Alberto D´Onofrio,	IEO	Milan	Italy
Dimitris Kafetzopoulos	FORTH	Crete	Greece
Juliusz Pukacki	PSNC	Poznan	Poland
Manolis Tsiknakis	FORTH	Crete	Greece

References

1) www.usability.net
2) www.iso.org
3) DIN EN ISO 13407 Benutzer-orientierte Gestaltung interaktiver Systeme (ISO 13407:1999); Deutsche Fassung EN ISO 13407,1999
4) DIN EN Part ISO 9241-11 Leitfaden Usability, DATech Deutsche Akkreditierungsstelle Technik GmbH, 2007
5) IEEE 830, Recommended practice for software requirements specifications, IEEE, Institute of Electrical and Electronic Engineers, 1998
6) Howden, W. (1980). Functional program testing, IEEE Transactions on Software Engineering 6: 162-169
7) IEEE 1016, Software design description,Institute of Electrical and Electronic Engineers, ANSI/IEEE Std 1016-1998, 1998
8) UML 1999, pp. 2.113- 2.123
9) Functional Requirements and Use Cases, www.bredemeyer.com
10) Andrew Dillon “Group dynamics meet cognition: combining socio-technical concepts and usability engineering in the design of information systems” (E. Coakes, (et al) (Eds.) The New Socio Tech: Graffitti on the Long Wall. London: Springer-Verlag, 2000, pp119-126
11) www.wikipedia.org

Tables

Figures

Abbreviations

ACGT	Advancing Clinico-Genomic Trials on Cancer
CAT	Custodix Anonymization Tool
CRF	Case report form
DATech	German Association for Accreditation GmbH
DICOM	Digital Imaging and Communications in Medicine
DIN EN	*DIN EN #* is used for the German edition of European standards
FhG	Fraunhofer Society – German research organization
GUI	Graphical User Interface
IAIS	FhG - Intelligent Analysis- and Information Systems
IEEE	Institute of Electrical and Electronics Engineers

ISO	International Organization for Standardization
ObTiMA	Ontology based trial management system for ACGT
QA	Quality Assurance
RDE	Remote data entry
SRS	Software Requirements Specification
UE	Usability Engineer

Glossary

Structural testing	Software structural testing means testing, verifying and evaluating the decisions made by the program using test cases based on the structure and logic of the design and source code. Complete structural testing implies the program's data structures (such as configuration tables) and its control and procedural logic at different test levels. The test levels are structured from the smallest module up to the complete product/ software. The three major steps are Unit-level structural testing, Integration-level structural Testing and System-level structural testing. Several structural test methods are available.
Traceability matrix	Traceability describes those attributes of the software that provide a thread from the requirements to the implementation with respect to the development and operational environment. The traceability matrix is a correlates any two baseline documents that require a many to many relationship to determine the completeness of the relationship. It is often used with high-level requirements and detailed requirements of the software product to the matching parts of high-level design, detailed design, test plan, and test cases. The measurement of requirements traceability is used to identify requirements that are either missing from, or are in addition to, the original requirements.
System interface description	Description of data types, data formats and methods used to allow two components of a software system to exchange information.