OPEN Project

STREP Project FP7-ICT-2007-1 N.216552

Open Pervasive Environments for migratory iNteractive services

[1]

Title of Document:	Usability criteria for project phases: use cases selection, design, development, test and deployment
Editor(s):	M. Stecca, S. Marzorati, A. Grasselli, M. Piunti
Affiliation(s):	Vodafone Omnitel NV
Contributor(s):	F.Paternò, C.Santoro
Affiliation(s):	CNR-ISTI
Date of Document:	November 2008
OPEN Document:	D6.1
Distribution:	INTERNAL
Keyword List:	Usability, Testing
Version:	Issued

OPEN Partners:

CNR-ISTI (Italy)
Aalborg University (Denmark)
Arcadia Design (Italy)
NEC (United Kingdom)
SAP AG (Germany)
Vodafone Omnitel NV (Italy)
Clausthal University (Germany)

[1]

Title: Usability criteria for project phases: use cases selection, design, development, test and deployment	**Id Number:** Errore. L'origine riferimento non è stata trovata.

Abstract

Scope of this document is to introduce the "Usability" concept, both in general and applied to the open migratory service platform (developed in OPEN project), and to individuate the concrete steps to follow during the design of the system, so that usability can be improved.

The application of methods necessary to perform usability testing will require interdisciplinary teamwork at different step during the product development lifecycle.

After a first part, where the process to adopt will be illustrated and the test plan will be described, organization of the usability testing is the main target of the deliverable, thus individuating in which moment of the project it is necessary to test and how each work teams should contribute.

A more specific definition of the test session will be described accurately in the deliverable D6.4- Testing & Validation Methodology.

Title:	Id Number: D6.1

Table of Contents

1. Purpose

The purpose of this deliverable is:

- To provide the general understanding of the “Usability” and “Usability Testing” concepts
- To explain the relationship between the usability and the product lifecycle
- To define the structure to adopt in the test plan
- To analyze the process necessary to improve the usability within the OPEN project, meaning:
    - To define during development when to execute specific usability tests
    - To individuate the end users and the general goal
    - To plan the process and the collaboration between partners in order to test the usability

2. Introduction

Generally, during product design and development, the attention is given to the overall machine or system, and often the needs of the end user are not taken into account. While in the past the technology was addressed only to expert users, today users are suitable to have some technical knowledge, and so it's becoming very important to consider, during the product design lifecycle, some aspects regarding the usability of the system from non technical people side.

Within this line, a variety of techniques, methods and practices, defined in usability engineering, can be applied at different points in the product development, but since applying all methods is expensive and does not ensure optimum results, we basically try to individuate the type of methods to use in each phase of the project. In fact, the usability engineering is not a one-shot affair fixing up the issues before the release of the product, but it requires a set of activities into the teamwork throughout the lifecycle of product.

Target of this deliverable is to define and organize the necessary approach to obtain the optimization of the system usability.

In order to have a better understanding of the "Usability" concept and its application during the lifecycle of the product, an overview will be laid out in the Chapter 2 and 3. Also, in these sections the specific goals of usability for the OPEN project will be individuated to reduce the analysis size and the timeframes during the development in which to test the usability. Since the OPEN project is subdivided in a programme, having as results a variety of documentation and vertical prototypes, the individuate phases of testing will be chosen within the overall timescale.

In the Chapter 4, it will be defined in detail how the test plan has to be found out and the necessary contribution of the work team participants to the OPEN project.

The detailed compilation of the tests plan will be developed in the following deliverable D6.4, while the testing results will be collected in the D6.5 and D6.7.

Therefore, the final document should be considered as task reference from all the work teams and the defined process will be integrated in development phase of the OPEN project.

3. Usability: an overview

Before dealing with the topic of usability testing, it is primary to understand the meaning of the *Usability* term.

The abstract concept of *Usability* can be defined through multiple precise and measurable components:

Learnability	The system should be easy to learn so that the user can rapidly start getting some work done through the system.
Efficiency	The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
Memorability	The system should be easy to remember, so that a casual user can be able to come back to the system, without having to learn everything from scratch.
Errors	The system should have a low error rate, so the users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Of course, catastrophic errors must not occur.
Satisfaction	The system should be pleasing to use, so that users are subjectively satisfied when using it.

Fig. 1: Usability concepts.

This is only a common used definition and there are other possible ones (from J.Nielsen, S.Krug, J.Rubin and D.Chisnell…), however they are very similar, and in any case we can see that usability can be not only a subjective matter of discussion, but it can be systematically approached, improved and evaluated, verifying the reaching of these goals:

- **Performance:** Time/steps required to complete primary actions (e.g. find something to buy, create a new account, and order the item.)
- **Accuracy**: Mistakes made by people and the effect of them (are they recoverable or fatal?)

- **Recall**: What does user remember afterwards or after periods of non-use?
- **Emotional response**: User perception/feeling about the tasks completed: Is the person confident/stressed? Would the user recommend this system to other?

The overall goal of *usability testing* is to update the design team by gathering useful data to identify and rectify lacks of usability existing in the product before releasing it. Really, the *usability testing* is not performed via formal numeric measurement studies of its attributes, but through evaluation methods focusing on users and their tasks, applied in an iterative design process during the lifecycle of the product. So, a structured and systematic approach is required in order to collect information from the users about specific product aspects.

Generally, the first step is to define the high-level goals and to analyze the end users and the tasks related to the planned product. Following, there is the necessity to individuate some people (*participants*) that can represent users on the design team itself.

Several methods can be used, based on some primitive tests models (cfr. Handbook of usability testing, J.Rubin and D.Chisnell)

- **Focus Group Research** model. Such model is applied in the methods at the very early stages of the project in order to define scenarios and requirements. All work teams employs the simultaneous involvement of more than one *participant*, that through storyboard on paper elaborate screen-based prototypes.
- **Survey** model. It helps to understand the preferences of a broad base of users about an existing or potential product. Survey has to show a clear language and to be understood similarly by all the possible readers. Surveys can be used at any time in the lifecycle but are usually used in the early stages to better understand the potential user.
- **Walk-Throughs** model. It is used to explore the user vision of a product through an early concept or prototype of the product. Usually the designer responsible for the work guides his or her colleagues through actual user tasks (sometimes even playing the role for the user), while another team member accounts difficulties faced or team's concerns.
- **Open and Closed Card Sorting** model. It is used to find out contents and functionalities with the help of the user. You can either give participants cards, showing content without titles or categories and have the users

naming, or give participants preliminary or pre-existing categories and ask them to sort content or functions into those.

- **Paper Prototyping** model. With this technique, an aspect of a product on paper is shown to the users, who are enquired about it. The questions can deal with particular attributes, such as organization and layout, or where to find certain options or info.
- **Heuristic Evaluations** model. A usability expert reviews the system according to accepted usability principles (Heuristics) from the body of research, human factors literature and previous professional experience.
- **Usability Testing** model. This model employs techniques to collect experience data while observing *participants* using the product and performing practical tasks. Testing is roughly divided into two main approaches. The first approach involves formal tests conducted as real experiments, in order to confirm or refuse specific hypotheses. The second approach, less formal but still rigorous, utilizes an iterative cycle of tests to expose lacks of usability and gradually shape/create the product.
- **Follow-Up Studies** model occurs after formal release of the product. The idea is to collect data for the next release, using surveys, interviews and observations.

Typically, these models are used in a combined/modified form, according to the project needs.

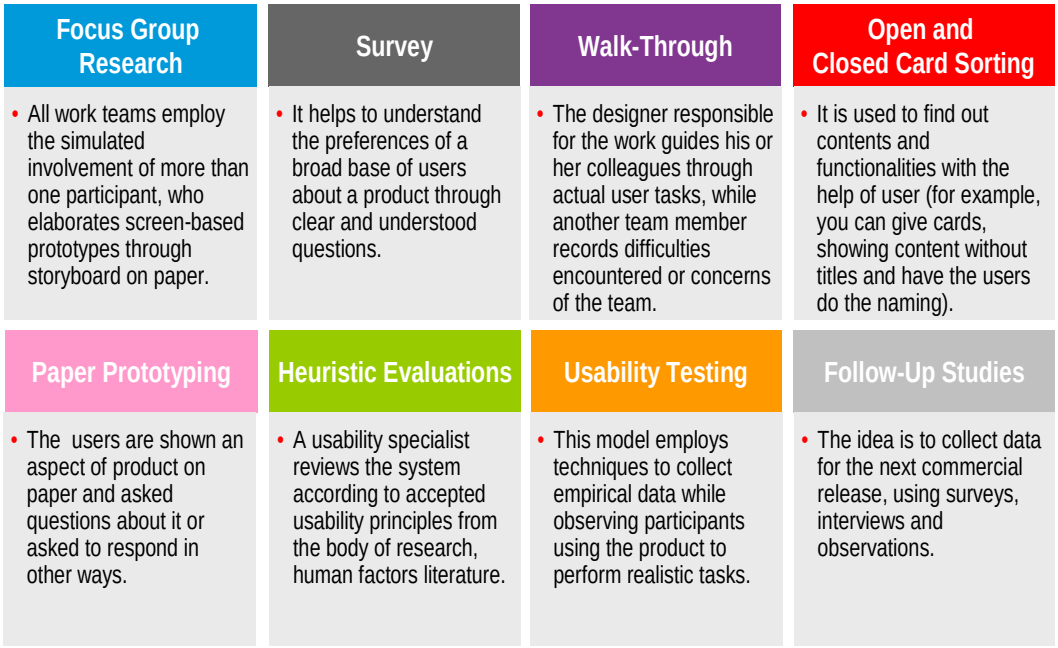

Fig. 2 Usability methods

.

3.1 General usability goals in the OPEN Project

In the OPEN project, the main usability focus concerns the migration aspect. In fact, since the scope of this platform is to provide the possibility for users to freely move, while continuing the interaction with the available applications through a variety of interactive devices, it will be necessary to verify that the transition from a device to another is:

- Easy to trigger from the user: user interface handling the migration has to be simple, clear and immediate
- Seamless and transparent when the user trigger is not needed
- Quick from user point of view, so to not interfere or limit the applications task
- Able to support a good graphical rendering
- Able to support dynamic user interface reconfiguration for the end user to satisfy the usability (not only when the devices involved in the migration adopt the same modality interaction, but also when the user can migrate changing the interface modality, e.g. from graphical to vocal)
- Able to manage in a practical way several type of migration: total, partial, distributing, aggregating, multiple migrations.
- Able to satisfy the needed of the user
- Satisfying the user for the impact on tasks
- Maintaining after the migration maintains the application usability characteristics provided before the migration.

In addition, the migration platform in order to be usable to provide:

- Easy configuration system in order to fix up some features of migration
- A good documentation or help about learning the use of migration system

Then, the methods to be used will be specified in order to assess the usability in the project. In the next chapter, it will be illustrated in more detail how is necessary to define in which stages of the project the tests will be set.

About the participants, their selection involves identifying and describing the relevant behaviour, skills and knowledge of the people who will use each product.

This system aims to allow the user migrating from a device to another one, without losing the state and with automatic adaptation of the contents to the features of new device. For the OPEN platform, there is not a single target user, but a wider range. In fact, keeping into account that the OPEN Project will apply

and analyze the system in two different reference domains, business and gaming, the end users should belong to this environment.

For the BUSINESS DOMAIN, the target user could be:

- Generally, a worker of medium age (25 to 45), keen on the technological use. The general use could involve both manual works for inspection, maintenance and emergency and office works.
- The user is a medium age person with technical skills, who wants to do shopping by different devices and synchronizes the list's shopping with the fridge screen, reporting the missing things.
- A business man that wants to participate to an auction, using several devices in different modalities (voice or screen).
- More business men finishing the work in the evening, so they can transfer the office files (PPT, mail, text, Word, etc) in several devices while watching TV and using communication services as IM.
- A business man in a phone conference, who transfers the call to his workstation, enabling him to see the other parties of the phone meeting.

Please note that business domain is not defined yet, so this is only a proposal to identify possible business users.

For the GAMING DOMAIN, the target users could be:

- Young students who love old fashion and sport games and like to associate gaming sport with real-time sport shown in TV. The users would like to move the game in different devices without losing the status of it.
- Adults, fan of web strategic turn based games, playing with different devices, migrating from each other.

4. The concept of Usability applied to Migration

The usability evaluation of migratory interfaces should consider their two main components: continuity and adaptation.

Since one of the goals of migratory applications is giving the user the feeling of no interruption during the task accomplishment, in order to define 'usable' a migratory application, first of all it has to support *continuity* in an effective way. In the context of migration, we can define continuity as the ease with which users are able to be aware that the same task will be continuously accomplished using (at least) two different devices. Therefore, it is highly important that the provided user interface on both devices (source device and target device) will effectively render this process in a seamless way. Therefore, we have to analyse the last user presentation with which the user interacts with on the source device, and the first presentation the user interacts with on the target device (see Figure below).

Indeed, on the one hand the user interface on the source device should "prepare" the users and make them aware of the fact that the interaction is about to continue on another device (in other words, an effective reference 'forward' should be provided to the user). In the context of migration, this means to consider the last presentation on the source device with which the user interacts with, which means considering the Migration Client.

On the other hand, the user interface on the target device should effectively render the fact that the presentation provided is a *continuation* of some work has already been done previously somewhere else (namely: on the source device) and that the provided presentation enable them to continue from that point onwards for accomplishing the same task using another device. Therefore, an effective reference 'backward' (what happened before) should be rendered on the target device. This is an important aspect since, if users do not recognise such a situation, they could be disoriented by the user interface presented on the target device.

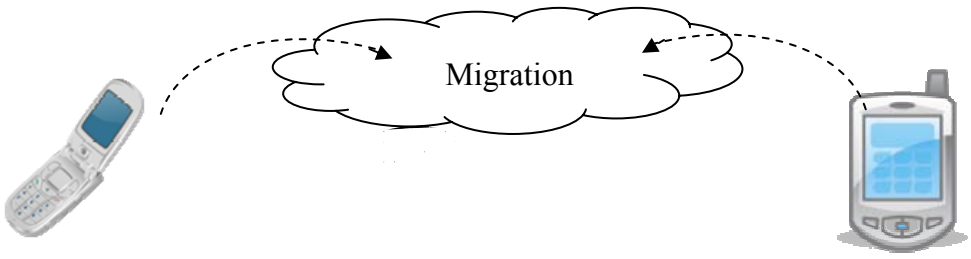

Fig.3 Supporting continuity in migration

4.1 The Migration Client

The migration client supports the selection of the target device and also decide which kind of migration has to be carried out (e.g.: partial/total, immediate/deferred).

- In case of *user-triggered migration*, the migration client should allow the user to e.g. understand precisely which devices are available for migration, to which physical devices in the current environment they correspond to, etc. and enable them to easily select what they judge to be the best parameters for migration (target device, type of migration, etc.).
- In case of *automatic migration* the migration client device should effectively present the characteristics of the suggested migration. Therefore, it should effectively render to the user e.g. the reason why a possible migration is suggested by the system (e.g.: a better device becomes available), the devices involved in the proposed migration, and so on.

In both cases, the Migration Client should be usable enough to make the users predict what their selection will result in migration terms. Therefore, the migration client should provide the user with the ability to understand the effect of triggering a certain migration. For instance, if a partial migration is selected, users should be able to easily predict what part of the user interface will migrate and on which device(s) the results of the interactions will appear. So, a factor that can impact the usability of migratory user interfaces is the *predictability* of the effects of the migration's trigger, namely the ability of the user to understand the effect of triggering a specific type of migration. To this regard, the migration client should be designed in such a way to effectively allow the users to understand the effects that a migration trigger will provoke on the target device(s). The predictability is connected with the number of different migration options the migration client will offer, and to what extent such options were designed in such a way to be able to effectively communicate to the user the result that will be achieved by activating each of them. This might be especially relevant when the migration process involves more than two devices (and/or more than one user), although different options can be also available with only one user and two devices involved (partial/total migration, immediate/deferred migration). For example, users should be able to easily understand to what real, physical devices in the environment correspond to the devices that can be selected as migration targets (suitable visualisation techniques should be foreseen to this regard).

4.2 The Adapted User Interface on the target device

When the interaction moves to the target device, users should easily understand that the user interface presented refers to an activity that already started before. This has to be recognised as easily as possible, and any disorientation issues

should be prevented in this phase: the adaptation strategy used to present the user interface on the target device is critical to this aspect.

Different factors can affect such recognition, and they might even make continuing the interaction in a seamless way a bit problematic from the user's point of view. Factors might include:

i) whether long *time* has passed since the last interaction, which therefore might be difficult to remember;

ii) whether the adaptation process has changed the user interface rendered on the target device in such a way that the users do not recognise that it enables them to logically continue the performance of their tasks from the point where they left off in the source device. For instance, the new user interface should clearly highlight the elements that were subject to change during the previous user's interaction with the source device. Due to screen size limitations, sometimes highlighting such information may not be immediately evident.

In the first case i), adequate feedback messages should be identified in order to take into account the aspects related to the time passed. For instance, if a long time has passed between the moment when the migration was triggered to the moment when the user starts to interact with the new interface on the new device, the adaptation engine should be able to adapt the presentation accordingly (for instance, by summarising to the user the changes that were performed in the source device). This should allow the users to more easily and more quickly contextualise the interaction and remember the action(s) already done on the user interface of the source device (e.g.: which sub-tasks they already accomplished on the source device, which overall task they were expected to complete, ..). Regarding the time factor, another aspect that can affect the user experience is the time necessary for the migration to take place: a migration that takes a long time to complete may compromise again the feeling of a continuous interaction and have a negative impact on the overall user's experience and satisfaction.

Regarding the second aspect ii), the adaptation process should be a trade-off between two sometimes conflicting aspects: on the one hand, since the involved devices are different, a user interface adapted to the changed interaction resources is needed; on the other hand, the adaptation process should not radically change the design, but trying to maintain a consistent logical interaction model in the new device, in order to make the users recognise a familiar interaction.

4.3 Other General Aspects

In the case of migration, *satisfaction* can mean carry out tasks in a more efficient and even pleasant way. Therefore, the system should be able to suggest migration to the users when conditions for more productiveness or more satisfaction occur in the environment. For instance, when migrating from a cellphone to a desktop platform, the possibility for having multiple activities concurrently visualised in

the same presentation will probably speed up the completion of the task when it is compared with navigating through a number of different presentations (as it will happen when using a platform with small screen size). In this case, a migration suggestion should be proposed to the user. In order to make the user appreciate the benefits of changing the interaction device for enabling a better interaction, it is important that the implications connected with the disruption caused by changing the interaction device are appropriately taken into account. For instance, we have to consider the time needed for performing the migration itself, the time needed by the user for adapting his/her model of the application to the features of the new platform, etc.: in order to have a satisfactory migration such amounts of time should be kept reasonably low. Finally, another aspect connected with usability of migration is *memorability* (especially useful for e.g. occasional users): the migration should be intuitive enough to enable the users to come back to the application after some time and be able to use it without having to learn its features all over again.

5. Usability during product Lifecycle

As hinted in the previous chapter, it is a good rule to start analyzing the *usability* of a product not only when the system is released, but during each phase of the development lifecycle.

So, the product is built in all its own parts taking into account the usability aspects and avoiding adapting the system only at the end, when it is not always simple and possible in satisfying mode. Moreover usability work made before the system is designed can allow avoiding to develop unnecessary features.

A summary of the lifecycle stages could be:

1. **Know the user:** to study the proposed users and usage of product
2. **Competitive analysis:** existing competing products are often the best prototypes we can get of our own product
3. **Goal Setting:** not all usability aspects can have the same weight, so it is necessary to make your priorities clear
4. **Parallel Design:** it is a good idea to start the design with a parallel process, in which several different designers work out preliminarily
5. **Participatory design:** designers should have access to a pool of representative users in order to interact with them during the project
6. **Coordinated design of the total interface:** phase in which the consistency of the project's interface, the documentation and the online help or tutorial are coordinated
7. **Apply guidelines and heuristic analysis:** stage in which specific guidelines are applied
8. **Prototyping:** stage in which true or paper prototypes for the final or partially system are created
9. **Empirical testing:** end user analysis of the product at current state
10. **Iterative design:** basing on the usability problems and opportunities disclosed by the empirical testing, a new version of the product can be released
11. **Collect feedback from field use:** collection of the usability data after the release of the product in the field

It's important to note that a usability engineering effort can be successful even if it does not include each possible refinement at all of the stages.

Direct contact between users and the design team is primary within the development lifecycle.

5.1 Usability Test during OPEN project lifecycle

In the OPEN project, we have individuated some timeframes, in correspondence of the production of some deliverables, where it is useful to realize a specific usability testing and we placed them according to the phases before described.

Since in the OPEN programme two iterations has been planned on processes and related results, some deliverables belonging to two cyclic periods will be distributed in the same phase.

Phase 1: Know the user

- D1.1: Requirements for OPEN Service Platform

Phase 2: Competitive analysis

- Not possible, because competing product are not available. A reference and a further analysis of the paragraph B 1.2.1 “State of the art of the DOW” and of the D7.2 could be made in the deliverable D6.5.

Phase 3: Goal setting

- D1.1: Requirements for OPEN Service Platform.

  (A global system goal setting has been already defined at the end of Chapter 2. Detailed goal will be defined for each test explicitly)

Phase 4: Parallel Design

- Approach not used.

Phase 5: Participatory Design

- They will be defined for each usability test in D6.4.

Phase 6: Coordinated design of total interface

- D2.4 Document about guidelines for multi-device user interfaces

Phase 7: Apply guidelines and heuristic analysis

Phase 8: Prototyping

- D1.3: Final requirements for OPEN Service Platform
- D2.1: Early Infrastructure for migratory interfaces
- D4.4: Migration service platform implementation
- D5.1: Initial application requirements and design (Definition of interfaces)
- D5.2: Initial prototype applications (Prototype)
- D5.3 Final application requirements and design (Prototype)

- D5.4: Final prototype applications (Prototype)

Phase 9: Empirical Testing

- Out of scope

Phase 10: Iterative Design

- Out of scope

Phase 11: Collect feedback from field use

- Out of scope

It should be clear from this list when the usability testing should be applied within the OPEN programme

The following step requires that the test procedure is found out for each period; so also the collaboration of own work teams, which better know the scope of the deliverable, is required. In the next chapter, it will be described the methodology of developing test procedures.

6. Test Plan

In order to define how each usability test has to be performed, the **test plan** has to be detailed.

For the OPEN project usability tests, the test plan will be compiled in the deliverable D6.4. The compilation of it will require the collaboration among leaders of WP6 and of the tested deliverable, as shown in Fig. 2.

In fact, the test plan is a document that should be matching the particular needs of the development team. So a first draft of the test plan will be produced and reviewed from all OPEN partners providing participants for the execution tests.

The test plan will be iteratively defined in stages according to the understanding (increasingly improving) of the test objectives by the people being involved.
Since the project in the OPEN programme is dynamic and could have some modifications during the development, when the tests are going to start, everyone who is directly affected by the test results should review the test plan.

The test plan also describes both the resources required to execute the test and the results to be collected in the deliverable D6.5 & D6.7.

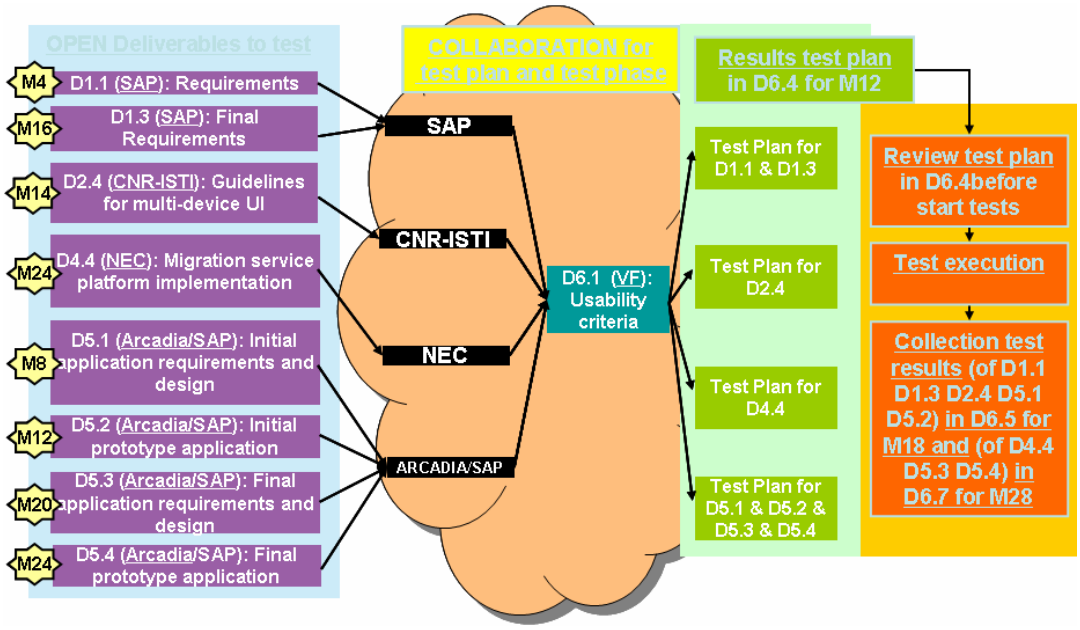

Fig. 2: Process in OPEN project for Usability Testing

The definition of each test plan and the test execution will be triggered by the deliverable issue.

As you can see from the Fig. 3, after the deliverable issue, VF-IT should elaborate a draft of test plan to compile in collaboration with the deliverable work team and to distribute to all OPEN partners for comments.

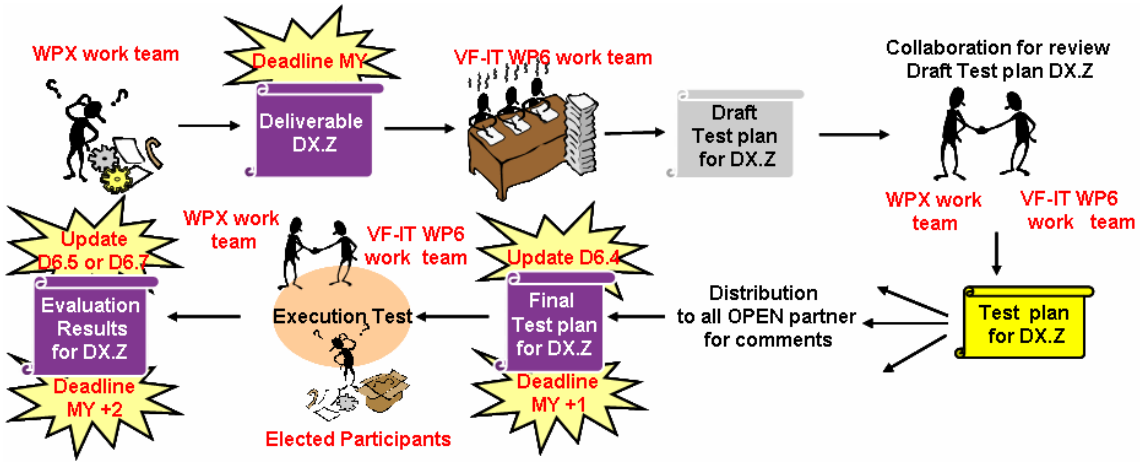

Fig. 3: Usability test process for each deliverable

After one month from the deliverable issue a **final test plan is product** and distributed as Internal Report, updating D6.4

Overall participants, VF-IT and deliverable work team, could **execute the test in around one month** and the results will be distributed as Internal Report, updating D6.5 or D6.7.

It is not possible to respect the fixed data for the D6.4 (M12), because for the layout out of the test plan it is necessary to know the contents of the deliverables related.

Also, it is important that all the partners are updated at anytime about the results (D6.5 and D6.7) and that these results are kept into account for the laid out of the following deliverables, so to solve eventual usability issues.

In the following table, for every deliverable considered in the usability testing process, it is indicated the OPEN partner providing collaboration to produce the draft test plan.

Title:	Id Number: D6.1

Deliverable	Collaboration for test plan
D1.1 Requirements	SAP
D1.3 Final Requirements	SAP
D2.1 Early Infrastructure for migratory interfaces	CNR-ISTI
D2.4 Guidelines for multi-device UI	CNR-ISTI
D4.4 Migration service platform implementation	NEC/CNR-ISTI/Clausthal
D5.1 Initial application requirements and design	Arcadia/SAP
D5.2 Initial prototype application	Arcadia/SAP
D5.3 Final application requirements and design	Arcadia/SAP
D5.4 Final prototype application	Arcadia/SAP

6.1 Usability Test during OPEN project lifecycle

Test plan formats could be different according to the type of test. However, a possible approach to follow is an adaptation to OPEN project environment of the "Handbook of usability testing", in which the typical sections to include are:

- Purpose, goals and objectives of the test
- Research questions
- Participant characteristics
- Method (test design)
- Task list
- Test environment, equipment and logistics
- Test moderator role
- Data to be collected and evaluation measures
- Report contents and presentation

In the following paragraphs a brief description of each section will be provided.

6.1.1 Purpose, goals and objectives of the test

This part of the document describes at high level the reasons to perform this kind of test at this time.

6.1.2 Research questions

It describes the usability issues and the matters needing to be resolved and focuses the research, as well as the rest of the activities associated with planning, designing, and conducting the test. It is essential that the research questions be as precise, accurate, clear, and measurable as possible. The research questions should originate from discussions with the development team or with individual developers, technical writers, marketing personnel and so on.

6.1.3 Participant characteristics

This section finds out the characteristics of the product end user. It is important both to define the type of users and to indicate the exact number of participant for type at the test.

For an effective experimental design, you must use a minimum of 10 to 12 participants per condition. However, for the purpose of conducting a less formal usability test, some researches have shown that 4 to 5 participants represent one audience cell that can expose about 80 percent of the lacks of usability of a product.

6.1.4 Method

This section describes how the test session will be performed. It should provide an overview of each side of the test, since the participants arrive until they leave, in enough detail so that someone observing the test will know roughly what he can expect. Examples of methods to be used are described in the Chapter 1.

6.1.5 Task list

The task list should consist of tasks that will ordinarily be performed during the utilization path of the product, such as documentation, and so on.

For the test plan, you need four main components for each task:

1. A brief description of the task
2. The materials and machine states required to perform it (document, scenario, mock-up, etc)
3. A description of successful completion of the task: criteria and measures
4. Timing or other benchmarks: time could be used as a criterion for success or a benchmark.

6.1.6 Test environment, equipment and logistics

This section of the test plan describes the environment you will attempt to simulate during the test and the equipment that the participants will require.

6.1.7 Test moderator role

This section helps to clarify what the test moderator (usually WP6 members) will be doing.

6.1.8 Data to be collected and evaluation measures

This section of the test plan provides an overview of the kind of measures to collect during the test, both performance and preference data.

Performance data, related to participant behaviour, includes error rates, number of accesses of the help by task, time to perform a task, and so on. Preference data, related to participant opinions or thought process, includes participant rankings, answers to questions, et cetera.

6.1.9 Report contents and presentation

This section provides a summary of the main sections of test report that will be included in D6.5 deliverable.

7. Conclusion

This deliverable has illustrated the usability evaluation procedures that we will adopt and that will be described in detail in D6.4, along with a brief discussion of what usability is and how to apply this concept to the OPEN Project,.

A specific choice of the OPEN programme's deliverables (during which to execute the tests) has been carried out and the guideline about the work organization has been laid out.

8. Reference

Morgan Kaufmann 1993: Jakob Nielsen *Usability Engineering*

Wiley Publishing, Inc. 2008: Jeffrey Rubin and Dana Chisnell: *Handbook of Usability Testing*

Denis, C., and Karsenty, L., Inter-Usability of Multi-Device Systems−A Conceptul Framework. In "*Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces*", H. Javahery and A. Seffah (Eds.), 373-385, Wiley and Sons, 2003.

Mori G., Paternò F., Santoro C. *Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions*. IEEE Transactions on Software Engineering August 2004, Vol. 30, No 8, IEEE Press, pp.507-520.