WIZUSE: A WEB-BASED WIZARD FOR SUPPORTING USABILITY TESTING PROCESSES AND PRACTICES

QING LI

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

DECEMBER 2005

Your file Votre référence
ISBN: 0-494-14328-2
Our file Notre référence
ISBN: 0-494-14328-2

ABSTRACT

WizUse: A Web-based Wizard for Supporting Usability
Testing Processes and Practices

Qing Li

Even if usability is becoming a hot topic both in industry and academic research, very little work has been done on the modeling and formalizing of empirical studies, in particular on usability testing processes. Process modeling is fundamental, not only for process maturity, standardization, and customization to different types of empirical studies, but also for the automation and integration of usability testing in the mainstream of software development lifecycle. Moreover, there is a big gap between usability testing practices and usability tools. Most of current tools used in usability testing cover only a few steps and activities of the testing process. We lack integrated toolsets that can support the whole usability testing process.

Within this context, we reviewed the existing usability testing process models while highlighting their limitations. We then proposed a ten-step, well-defined and structured process that combines all strengths and overcomes the drawbacks of current processes. We also demonstrated how such a well-defined process can be embedded into a web-based wizard, WizUse, to assist usability professionals managing, customizing, and conducting usability tests. To validate our approach, we also conducted a series of tests to demonstrate that the proposed process model and WizUse are a suitable approach to bridge the gap between current usability practices and tools.

Keywords: Usability, Usability testing, Process, Tools, Methods, Wizard, WizUse.

ACKNOWLEDGEMENTS

The help and support of a number of people made this thesis a reality, and I am grateful to each of them:

Dr. Ahmed Seffah for his guidance and patience throughout my Master's period. He gave me the opportunity to participate in a number of challenging and interesting projects as a member of the HCSE (Human-Centered Software Engineering) Group at Concordia University.

Naouel Moha and Armand Nganou, my team-mates of RANA project, for their hard working, unselfish help, and contribution to the RANA project.

Dr. Xiao-Wen Chang, my mentor and best friend, who kept an eye on the progress of my work and was always available when I needed his advice.

Tiefei Liu, Yuhui Xia, Hanzhen Zhang, Chenghai Zhang for their support and encouragement.

Anand Joshi, Mohamed Taleb, Yojana Joshi, Daniel Sinnig, Homa Javahery, Muhammad Faraz Anwar, and Harkirat Kaur Padda for their help of reviewing this thesis and giving great feedback.

Above all, I am grateful to my parents, for their active support, for believing in me, for putting up with me, and for unselfish love sharing both happiness and misery with me!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AFIHM:	Francophone Human-Computer Interaction Association
A/V:	Audio/Video
API:	Application Programming Interface
CAUTE:	Computer Aided Usability Testing Environment
CASE:	Computer Aided Software Engineering
CIF:	Common Industry Format
CSS:	Cascading Style Sheets
ER:	Entity Relationship
HCI:	Human Centred Interaction
HCSE:	Human Centered Software Engineering
HMD:	Head Mounted Display
HTML:	HyperText Markup Language
HTTP:	HyperText Transfer Protocol
IEC:	International Electrotechnical Commission
ISO:	International Standardization Organization
LCD:	Liquid Crystal Display
MOUDIL:	Montreal Online Usability Digital Library
MPEG:	Moving Picture Experts Group
MVC:	Model View Controller
NIST:	National Institute of Standards and Technology
OOSE:	Object Oriented Software Engineering
PDA:	Personal Digital Assistant
QUIM:	Quality in Use Integrated Measurement
RANA:	Remote Architecture for Net-Based Analysis
SIGCHI:	Special Interest Group on Computer-Human Interaction
SPSS:	Statistical Package for the Social Sciences
STC:	Software Technical Communication
UCD:	User Centered Design

UI:	User Interface
UPA:	Usability Professionals' Association
XML:	eXtensible Markup Language

1. Introduction

1.1. The Concept of Usability

1.1.1. Usability Definitions

Usability has been around since the early seventeen century. Historically, it has been defined in multiple ways [Dillon 2001]:

- Semantics: in this case usability is equated to terms such as "ease of use" or "user-friendliness", without formal definition of the properties of the construct.
- Features: here, usability is equated to the presence or absence of certain features in the user interface such as Windows, Icons, Menus or Pointing devices.
- Operations: where the term is defined in terms of performance and affective levels manifest by users for certain task and environmental scenarios. All presented usability definitions below belong to this level.

Nielsen [1994b] advanced that usability is not a single, one dimensional property of a user interface, but it should associated with learnability, efficiency, memorability, errors and satisfaction that also influence the overall usability of a system.

Under the auspice of the International Standardization Organization (ISO), a series of standards related to usability have been developed. The most representative ones are the ISO 9129-1 and ISO 9241-11. In the ISO 9126-1, the definition of usability is depicted as "The capability of a software product to be understood, learnt, used and attractive by a set of users under specified conditions" [ISO/IEC 9126-1, 2001]. The ISO 9241-11, Ergonomic requirements for office work with visual display terminals, defines usability as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [ISO 9241-11, 1998].

Usability is a big umbrella since it has a large space in which we can define its meanings. To a certain extent, its definitions are very vague in terms of having both generic and specific meanings. In addition, the meaning of usability will probably change according to different context and perspectives. In this thesis, usability is looked as the external attributes of software, mainly focusing on user interfaces of computer software applications and websites. The detail information related to this concept and its evolution will be present in chapter 2.

1.1.2. Usability Benefits

Throughout the last decade, a large amount of cases studies [e.g. Bevan 1999, 2000;Bias and Mayhew 1994;Donahue 2001;Ehrlich 1994;Karat 1994], has been conducted and many efforts have been spent in industry on the importance of usability and its engineering. An increasing number of people agree with the idea that usability is beneficial to both end users and commercial software companies.

There are two major benefits to consider usability when developing software applications: reducing costs and increasing sales. The following table shows the sub-items that are composed within these two benefits. [Donahue 2001]

Reducing Costs	Increasing Sales
▪ Improved productivity and efficiency ▪ Reduced errors ▪ Reduced development costs ▪ Lower maintenance and support costs ▪ Reduced training costs ▪ Increased job satisfaction	▪ Increased customer satisfaction ▪ Improved competitive edge ▪ Increased market share ▪ Product/service differentiation ▪ Advertising advantage ▪ Better notices in the media

Table 1-1: Benefits of Usability

A number of business cases demonstrate also that an early focus on usability has a win-win consequence for both end-users and companies who develop the software. For example, "Wixon & Jones carried out a case study of a usability-engineered software product that increased revenue by more than 80% over the first release of the product (built without usability work) [1992]. The revenues of the usability-enhanced system were 60% higher than projected. Many customers cited usability as a key factor in buying the new system." [Bias and Mayhew 1994]

1.1.3. Usability Engineering

The term Usability Engineering was first used by usability professionals at Digital Equipment Corporation [Good et al. 1986]. The term was used to denote concepts and techniques for planning, achieving and verifying the usability of a system. Mayhew defined Usability Engineering as a discipline that provides structured methods for achieving the usability of a user interface design during the product development [Mayhew 1999]. Several disciplines are the bases for usability engineering including cognitive psychology, experimental psychology, ethnography and software engineering.

Historically, usability engineering techniques aimed to provide usable user interfaces including effective interactive presentations of both information and functionalities. Recently, it expanded to other software development activities, particularly requirements analysis and system envisionment [Rosson and Carroll 2002]. More and more, usability engineering is considered throughout the whole software development process and to determine what user interfaces and functionalities are necessary, as well as how they should be effectively presented. Usability can be applied as early as to requirements analysis by defining usability goals and as late as alpha test of applications to verify the degree to which the software met the usability requirements or goals having been set.

1.2. Usability Evaluation and Testing

Usability evaluation and testing methods form the core of usability engineering. There are three distinct categories of usability evaluation methods [Dillon 2001]:

- User-based: where a sample of the intended users tries to use the application.
- Expert-based: where an HCI or usability expert makes an assessment of the application.

- Model-based: where an HCI expert employs formal methods to predict one or more criteria that quantify the user performance.

1.2.1. User-Based Methods

Usability testing is the most representative method in this category. It consists of testing an application with a sample of users performing a set of pre-determined tasks [Dillon 2001]. After test participants finish their tasks, they are often asked to attend an interview session to debrief their performance during the test. They also are asked to fill some surveys or questionnaires providing their likes and dislikes about the application. Usability tests can be performed either in a usability test laboratory or in the real work environment site. Along with conducting of usability tests, concurrent verbal protocols, such as thinking aloud protocol, might be solicited to shed light on users' thought while they are interacting with the application so that issues of comprehension and user cognition can be addressed. [Dillon 2001]

There are many variant methods of user-based tests, some of which are more informal. For example, some user-based tests are unstructured, involving the user and the usability professionals jointly interacting with the system to gain agreement on what works and what is problematic with the design. [Dillon 2001].

1.2.2. Expert-Based Methods

Expert-based methods refers to any form of usability evaluation which involves human computer interaction (HCI) expert, domain experts of the target application or both, examining the application and estimating its likely usability for a given user population. Users or participants are not involved in such cases. Usability evaluations depend only on the judgments and interpretations of evaluators and their expertise in usability engineering.

Two expert-based usability evaluation methods are largely used: Heuristic Evaluation [e.g. Nielsen 1994], and Cognitive Walkthrough [Wharton et al. 1994]. Both of them aim to examine and report usability problems with a user interface.

1.2.3. Model-Based Methods

Several methods are proposed in the model-based method category, such as the most famous one, GOMS model. These methods can accurately predict certain aspects of user performance with an interface such as time to task completion or difficulty of learning a task sequence. In such cases, the evaluator determines the exact sequence of behaviors a user will exhibit through detailed task analysis, applies an analytical model to this sequence and calculates the index of usability. [Dillon 2001]

1.2.4. When to Apply Usability Evaluations

Whatever usability evaluation and testing method is used, the purpose of any evaluation is to derive measures of effectiveness, efficiency and satisfaction of an application or website, identify usability problems, and provide redesign recommendations to software development team to improve the usability of the application.

Usability evaluation can be considered at any stage of the software development lifecycle. In the requirements analysis stage, usability can be refined by usability expert evaluations or feedback from target users of the application. In this stage, model-based evaluation methods can be used to predict the usability of an application before the application is actually designed and implemented. In design and implementation stages, several user-based and expert-based evaluations can be conducted to determine usability of the application and to identify usability problems. Redesign recommendations can be provided to the development team as the inputs for the next iteration of design or implementation. After implementation of the software, an intensive usability testing can be conducted to verify and assess formally the usability. By doing so, problems can be identified and usability can be improved before software release. Usability evaluation can even be done after software release. The evaluation's findings can benefit the next version of the software or related products.

Getting user representatives to be involved in the software development process and conducting usability testing as early as possible can uncover usability problems at an early stage before they are deeply embedded, and thus avoid major overheads on reworks down the line. Evaluate before either detailed user interface design or development have begun minimizes costly revisions to completed design or code [Mayhew 1999]. This up-front investment will pay off in the long run. In the world of software design, it is a common lesson that the costs of changes grow exponentially throughout the development process. "It is a rule of thumb that the cost of fixing a problem is 10 times higher during product development, than in the period of prototyping, and 100 times as high after product release." [Olsen 2002] In a word, to answer the question when to apply usability testing, the definite solution is: "the earlier and the more iteratively, the better".

1.3. Motivation

Although the large variety of usability testing and evaluation methods exists today, the underlying usability testing processes are incomplete, not well-defined and informal. Currently, there are four usability testing processes, including Nielsen usability evaluation process, Rubin testing process, C&C usability testing process and Mayhew's usability engineering lifecycle process. All these four processes will be discussed in chapter 2. Their drawback in terms of formality will be highlighted. In process modeling, formalities provide a clear answer to the questions "what to do", "when", "who can do it", and "how". These formalities are crucial because when usability testers, especially novice ones, want to apply a usability testing process, they need clear ideas about the sequence of steps and activities that are going to conduct, which actors require to involve and what usability methods and tools they can use.

We are lacking the formalities in current usability testing processes, such as clearly defined actors and their roles at each step of the process. Moreover, we lack usability tools that can assist usability professionals to conduct tests throughout the whole process including the automation of repetitive activities.

Currently, thanks to the Internet and related technologies, usability professionals can conduct usability tests remotely. Remote usability tests are especially useful for evaluating the usability of web-based applications and web sites. However, it raises another problem that there is no usability testing infrastructure that fits remote usability tests. How to plan and manage usability tests remotely is really a challenge.

Overcoming the challenges addressed above is the major motivation for the research related to this thesis.

1.4. Objectives and Scope of Thesis

The objectives of this thesis are to define the usability testing process more formally as well as to propose a tool that can assist professionals in customizing and conducting tests. The major milestones are:

1. Collect and gather the expertise and experiences about usability tests from usability professionals in industry.
2. Propose a new usability testing process that aims to provide a well-defined and structured process model, which combines the major strengths, and overcomes drawbacks of current processes and practices.
3. Define, design and implement a web-based wizard, the WizUse, to assist usability professionals managing and conducting usability tests. WizUse aims to visualize the usability testing process while combining usability tools, templates, methods, and actors together. All components can be accessed through a unique interface via Internet.
4. Validate the WizUse by conducting usability tests, and in particular, remote usability tests

My research is part of the RANA project that will be presented in Chapter 4. The implementation of the RANA platform is divided into 3 phases:

- Phase one – Design a general purpose usability testing process.
- Phase two – Design an architecture to support remote usability testing; design and implement a web-based wizard to assist usability professionals managing and conducting usability tests.
- Phase three – Integration of usability tools such as QUIM Editor, MOUDIL, etc. Detailed descriptions of these tools are provided in appendix B.

The thesis covers phase one and two of the RANA project, which aims to propose a new usability testing process and develop a web-based tool, WizUse, which can:

- Customize usability testing processes;
- Visualize the usability testing process;
- Assist usability professionals to accomplish their usability tests;
- And automate the testing process.

In particular, first, I will propose a new usability testing process based on HCI standards, existing usability testing processes as well as best practices and experiences in industry. Second, I will introduce the Remote Architecture for Net-Based Analysis (RANA) project [Moha et al. 2004a], which aims to propose a Computer Aided Usability Testing Environment (CAUTE) to support remote usability testing. I will also discuss the architecture that I designed for remote usability testing. Third, I will present the design and implementation of WizUse, which is a component tool of RANA. Finally, I will present all validation efforts that have been done related to the proposed usability testing process and the WizUse.

1.5. Research Methodology

Our research methodology includes the following activities:

1. A literature review and a critical evaluation of the current usability testing processes and tools. This is the theoretical foundation of my research.
2. Propose a usability testing process, within which the activities, the actors, the usability methods and tools as well as the outputs involved in each step are clearly defined.
3. Conducted two surveys. One is related to the usability testing process and practices, and the other one is related to usability tools. In addition, several usability professionals were invited to do some expert evaluation to validate the proposed process.
4. Designed and implemented WizUse.

5. During the design and implementation of WizUse, some expert evaluations have been done to improve the application.
6. Conducted a series of usability tests for a head mounted display (HMD) system with the support of WizUse in order to validate it.

1.6. Organization of the Thesis

The organization of the thesis is as follows:

This chapter 1 presents the motivations, objectives and research methodology of this thesis.

Chapter 2 consists of a literature review that summarizes and critically evaluates the **"Related Work on Usability Testing Process"**. First, I provide some definitions for common terms and background relevant to this thesis. Second, I present four of the current and most representative usability testing processes. I discuss both their advantages and their drawbacks. Lastly, I further discuss the problems that current usability testing processes may have.

Chapter 3, **"The Proposed Usability Testing Process"**, proposes a usability testing process that aims to provide a well-defined and structured process to combine all the strengths and overcome drawbacks of current processes.

Chapter 4, **"WizUse: Design and Implementation"**, introduces the RANA project and the architecture for remote usability testing that I designed. It also gives some background information about the wizard and its design. Finally, it presents how WizUse, a web-based wizard for supporting usability testing, is designed and implemented.

Chapter 5, **"Validation"**, presents all the efforts that have been made related to validate the proposed usability testing process and WizUse.

In the last chapter, **Chapter 6 "Conclusion and Future Work"**, I am going to summarize my work and the future research that can be done.

2. Related Works on Usability Testing Process

In this chapter, firstly, the relevant background information related to usability engineering is presented. Secondly, the literature review related to usability testing processes, usability tools and methods is summarized. Four most well known processes are discussed, and their weaknesses are also presented. In the last part of this chapter, a further discussion about problems of current usability testing processes is given.

2.1. Terms Definitions and Background

2.1.1. What is Usability?

Perhaps the most well known definition of usability is the one given by Nielsen [1994b]: "usability is not a single, one dimensional property of a user interface, but is traditionally associated with learnability, efficiency, memorability, errors and satisfaction." Nielsen uses these five attributes to measure usability, which can be decomposed into more precise and measurable metrics or measurements. The focus of this definition is mainly on the user interface of computer software applications and websites.

Likewise, in the past two decades, the International Standardization Organization (ISO) also developed a set of standards that are related to usability, providing not only the definitions and principles, but also the process and measurements to achieve usability. These standards can be classified into the following four categories [Abran et al. 2003]:

1. Product effect (output, effectiveness and satisfaction at the time of product using);
2. Product attributes (interface and interaction);
3. Process used to develop the product;
4. Organization's Capability.

The following figure depicts the classification of ISO standards related to usability in terms of the four groups mentioned above. [Bevan 1999]

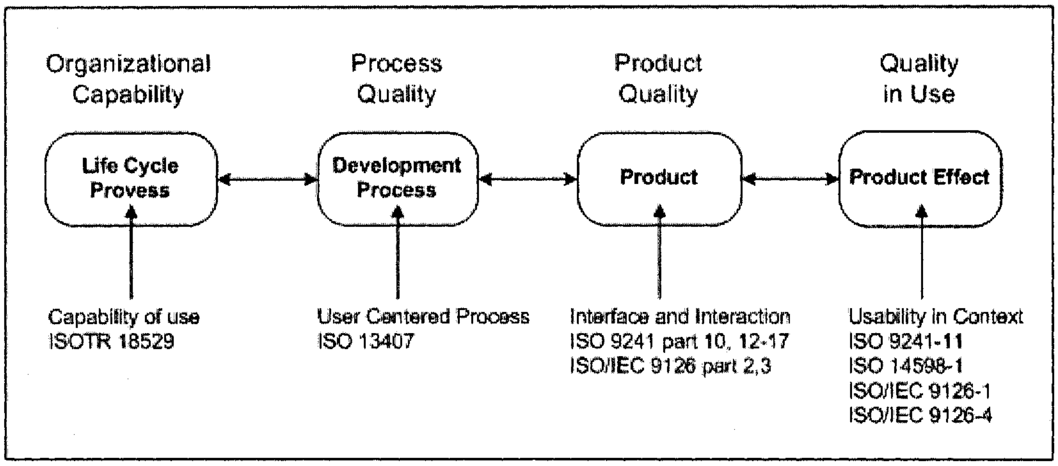


Figure 2-1: Classification of ISO Standards Related to Usability

Amongst these ISO standards, I only focus on the ISO 9241-11 and ISO 9126 standards, which are most frequently used when addressing usability.

ISO 9241-11: Guidance on Usability defines usability as high-level quality objectives in terms of measurable design attributes, and the extent to which the product meets the needs of the user. This definition defined usability in a broad and implicit way [Bevan 2001;Seffah 2003b]. The usability of a product is described as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [ISO 9241-11, 1998]. The definition in ISO 9241-11 gives the general conceptual guidance and principles rather than detailed specifications. Usability is considered as the final goal for the product that can meet the user's needs.

Meanwhile, ISO/IEC 9126: "Software product evaluation – Quality characteristics and guidelines for their use" considers usability as "a set of attributes that bear on the effort needed for use, and in the individual assessment of such use, by a stated or implied set of users." [ISO/IEC 9126, 1991]. ISO/IEC 9126 has been replaced by a new four-parts standard. The first part is Software Engineering – Product quality. In this part, a new software quality model is raised with a new term of "quality in use", which is defined as "the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use." [ISO/IEC 9126-1, 2001]. The following figure demonstrates the structure of the software quality model under the concept of quality in use. [King 2003]

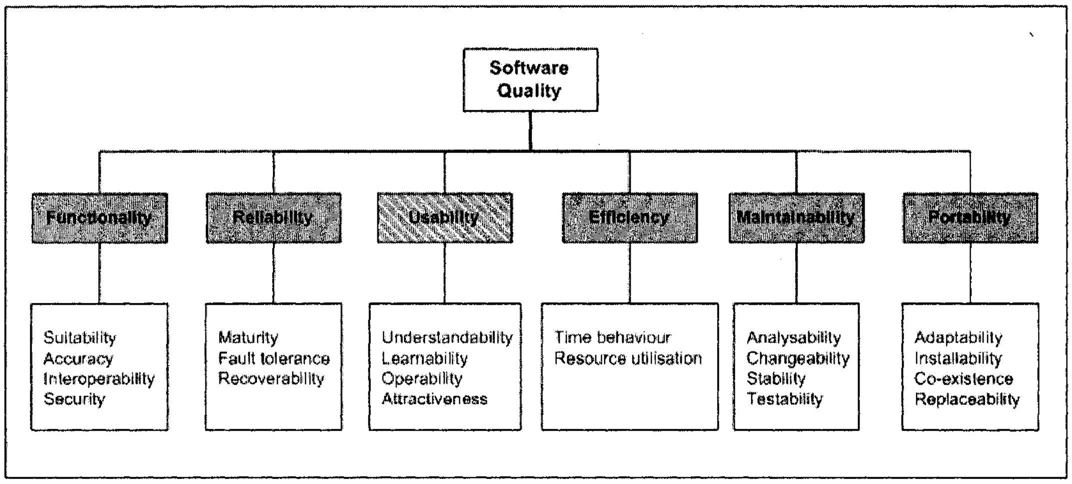


Figure 2-2: Software Quality – ISO 9126

As illustrated in the figure, quality in use has four high level factors: effectiveness, productivity, safety and satisfaction, which are decomposed into six software quality categories that are relevant in the software development lifecycle. Within this quality model, usability is one of these six categories, meaning that usability is one of the important aspects in terms of software quality. The definition of usability is depicted as "the capability of the software product to be understood, learned, used and attractive to the user, when user under specified conditions" [ISO/IEC 9126-1, 2001]. The purpose of the ISO/IEC 9126 [2001] standard is to provide a framework of the evaluation of software of evaluation [Abran et al. 2003]. Compared with the ISO 9241-11 standard, the definition of usability in ISO/IEC 9126-1 is more specific in which it provided detailed attributes.

Three different definitions of usability have been addressed above while many others coexist. In spite of different interpretations, the key points of these definitions are the same. These key points help us to have a clear overview of usability. First, usability is firmly connected to the context of use. We cannot discuss or measure usability without putting the software application in a certain context. Context of use includes four aspects:

Users: for whom the application is designed. In this aspect, the user's knowledge, skills, education and physical and mental attributes are very important when we are considering the usability of a product.

Tasks: the activities that users take on to accomplish a certain goal. Task duration and frequency are two key characters that should be taken into account. In addition, the sequence of steps involved in conducting a task is also very crucial.

Equipment: the software and hardware platforms the application is running on. Examples are: the target operating system and the minimum RAM and hard disk space requirements.

Environment: the aspect is related to the physical and social environment, within which the application is running, for instance, the color and the light intensity of background.

Secondly, there are three different perspectives when we look at usability: user view, developer view and stakeholder view.

- ***User view:*** users are mainly interested in using and interacting with software applications. They only focus on observable external attributes of usability, such as how the application meets their needs and how efficient the application can be used. Users do not care about the internal quality of the software. Most usability issues are related to user interface.

- ***Developer view:*** developers are more interested in internal attributes of usability. For example, what data structure serves better in terms of maintainability? Their core usability objective is to design and implement a useful application to meet the user's requirements.

- ***Stakeholder view:*** stakeholders mostly care about the overall usability of a software product. They consider usability as a high-level goal of the product that can provide services to users in an effective, efficient way. Users' satisfaction is most important to them.

In summary, usability of the software is composed of internal, external and general attributes, which are all related to ease of use and ease of learning within a certain context.

In order to avoid the confusion, the concept of usability in this thesis is defined as external attributes of software, mainly focusing on the user interfaces of computer software applications and websites.

Usability can be achieved by many usability methods. Dumas & Redish [1999] proposed six methods to achieve usability that can be carried out throughout the lifecycle of a product's development:

- Engineering it into a product through an iterative design and development process,
- Involving users throughout the process,
- Allowing usability and users' needs to drive decisions,
- Working in teams that include skilled usability specialists, interface designers and technical communicators,
- Setting quantitative usability goals early in the process,
- Iterating the design.

However, the most efficient method is usability testing, which will be detailed in the next section.

2.1.2. What is Usability Testing?

Usability testing or usability test is a method by which a certain number of participants, which are selected from the potential users of a product, are asked to perform certain tasks in an effort to measure and collect usability data according to predefined usability metrics, such as task performance time, percentage of tasks' failure or accomplishment [Dumas and Redish 1999;Mayhew 1999;Rubin 1994]. After analyzing the collected usability data, usability specialists abstract usability patterns and findings. Based on these patterns and findings, a final report will be generated, indicating usability problems and providing modification recommendations to improve application usability.

Unfortunately, the term "usability testing" is often used in a generic as well as a specific sense [Dillon 2003].

In the generic sense, it is used as a descriptive term to represent all kinds of usability evaluations, including heuristic evaluations, expert reviews and cognitive walkthrough.

In the specific perspective, it is used to denote the type of participants involved in usability evaluation. In this thesis, for the sake of consistency and simplicity, the term usability test or usability testing is understood from a rather specific perspective and denotes the evaluation procedure based mainly on user or participant involvement.

Usability testing can be classified into three different types [Rubin 1994]:

1. ***Diagnostic usability testing***: main purpose of this kind of test is to diagnose both usability flaws and good designs during software development, so that the development team can maintain the good designs and fix usability flaws.
2. ***Comparative usability testing***: comparison is very effective when there is more than one design or competitor's software in the market. One can also compare different versions of a software application and websites. After the comparison, advantages and disadvantages can be identified, upon which the development team can make the decision.
3. ***Verification usability testing***: at the end of development, the usability goals need to be verified, to show that all is well or to show how close you have come to preset usability goals.

All these three types of usability testing are dedicated to an iterative product development lifecycle, but in different stages. As showing in the Figure 2-3 [Rubin 1994], comparative testing can be employed at anytime from requirements analysis to product building, while diagnostic testing can be applied in the stages 2 to 4 of the development lifecycle. Verification usability testing is conducted at the end of the development.

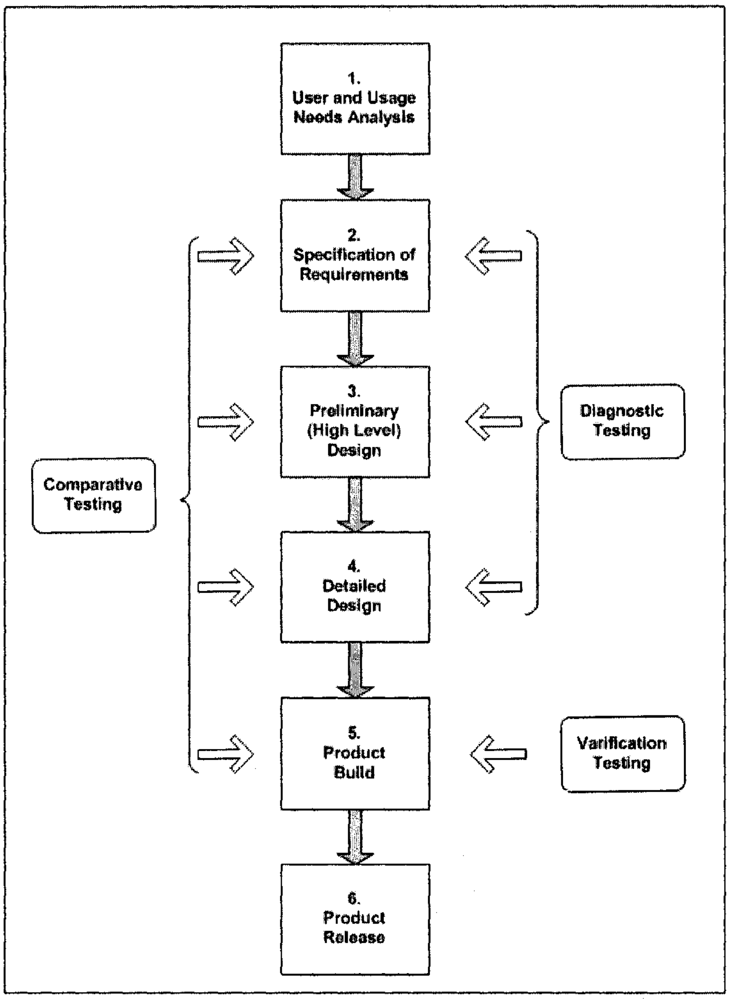


Figure 2-3: Product Development Lifecycle

2.1.3. Data Collection and Measure

During usability testing, several qualitative and quantitative data are collected from participants by usability testers. There are several perspectives when we want to differentiate data types. The data can be behavior or opinion, quantitative or qualitative [Hodgson 2003]. Usability testing is primarily about behavioral data, for instance, what the user's performance was, and when he/she interacted with the system. The behavioral aspect makes usability testing different from other techniques, such as interviews and focus groups. At the same time, usability testing also considers opinions and comments from participants when they interact with an application or a website. They always give preferences based on their characteristics, for example, opinions like "I like the color scheme of this design", or "these icons are too small to be recognized."

Mostly used classification criteria are qualitative and quantitative. Qualitative data can be the notes of what the participant did and said, or may be the comments from test observers. Quantitative data includes various aspects of users' behavior, such as:

- Time to complete a task,
- Number of errors or problems in completing the task,
- Number of requests for assistance.

We can perform statistic analysis on quantitative data to get some interesting findings. Which kind of data is more important for a usability test, qualitative or quantitative one, depends on the types of the testing. When the goal is comparison or verification, we prefer to have more quantitative data. For diagnostic testing, quantitative data may be less important than a qualitative list of problems.

Usability testing can be done formally, in a usability lab with audio and video equipments, or informally in which a paper-based prototypes of an application or web site are discussed with the user. Modifications are made to the application or website based on the usability tests' results. Whether the test is formal or informal, concurrent verbal protocols, such as thinking aloud protocol, might be solicited to shed light on users' thought while they interacting with the application so that issues of comprehension and user cognition can be addressed. [Dillon 2001]

2.2. Usability Testing vs. System Testing

A user-involved usability testing is an empirical study that aims to prove or ensure that the software is 'user problem free', as distinct from testing the functionality of the software, which is a test to prove or ensure that the software is bug free. Table 2 summarizes the goals of user-involved usability testing, and meanwhile highlights the major differences from traditional system tests (also see [Nielsen 1994b;Seffah and Metzker 2004a] for detailed discussion on usability testing).

Usability Testing	System Testing
Finding usability problems mainly with the user interface.	Finding defects, errors, and bugs in programs.
Tests can be done using a low-fidelity prototype or the system after its development and deployment.	Tests are conducted using a fully functional system.
Different levels of testing (e.g. task, behavioural, validation).	Different levels of testing (e.g., class, data, program, application, module, etc.).

Table 2-1: System Testing vs. Usability Testing

Roughly, in a usability testing, the users' thoughts, expectations, and behavioural reactions are observed, their commentaries and actions are recorded, and all these information are analyzed with the purpose to:

- Find any users' problem, model users' behaviour and quantify their experiences with the software system;
- Assess the usability, efficiency, effectiveness, and accessibility of a fully functional interactive systems or a prototype [Zhang et al. 1999];
- Measure user performance, productivity, and satisfaction [Rubin 1994];
- Compare different versions of the same product or similar products from the user perspective [Nielsen 1994b]; and
- Establish benchmarks over a longer period of time to avoid user problems, to understand competitors, and to produce guidelines for features, user interfaces, and interaction design.

2.3. Usability Testing vs. Heuristic Evaluation

Usability testing is one of the most important usability evaluation methods, besides which, a variety of other usability evaluation methods have been developed in the last ten years. Heuristic evaluation is the most popular one. Heuristic evaluation is performed as a systematic inspection of a user interface design for usability according to a set of established guidelines or principles (the “heuristics”) [Nielsen 1994b;Nielsen and Molich 1990]. The purpose of heuristic evaluation involves “examining the interface and judging its compliance with recognized usability principles.” [Nielsen 1994b]

When compared usability testing with heuristic evaluation, many similarities can be found. For example, the goals of these two methods are the same. They are all used to find the usability problems in the software design and give recommendations to the development team to improve during the next development iteration. For this reason, they are a corporate stage in an iterative design process. To some extent, usability testing can be regarded as a special kind of heuristic evaluation. When end-users interact with an application, they more or less perform some kind of heuristic evaluation on the basis of their own intuition and common sense instead of guidelines or principles.

However, there are many existing differences between usability testing and heuristic evaluation, which make them distinguish from each other. Their main differences are summarized in the following table.

Usability Testing	Heuristic Evaluation
Participants	
Selected representative users from target audience pool.	Human factors specialists, usability experts, or domain experts.
Who conduct the tests?	
Team of usability professionals	People who do the usability evaluation.
Testing focus	
The properties of the interaction between a representative user and the product.	Interface functionalities and design of the product.
Tasks performed	
A set of tasks according to predefined scenarios.	Overall inspection of the product, no specific task is performed.

Who generate the test report?	
Usability testers.	Expert evaluators

Table 2-2: Usability Testing vs. Heuristic Evaluation

Heuristic evaluation is especially valuable due to limited time and resources. A relatively high-quality output, including a report of findings and recommendations, can be generated in few weeks since the method does not involve the detailed scripting or time-consuming participant recruiting of usability testing [Kantner and Rosenbaum 1997]. On the other hand, because expert evaluators are no real users, sometimes the results of expert evaluations are irrelevant from the user's perspective. In other words, it is not accurate enough [Law and Hvannberg 2002]. In summary, usability testing is valuable when making clear-cut design decisions about products or services. It collects first-hand user data, and more accurately reflects the experiences of the product intended audiences.

2.4. Usability Testing Environments

There are many environments to conduct usability testing. In terms of places in which the tests are conducted, three major categories can be identified:

- Local usability testing
- On-site usability testing
- Remote usability testing

2.4.1. Local Usability Testing

Local usability tests are more often used to conducted formal usability tests with high-fidelity applications. Participants, selected from the potential users according to the predefined persona criteria, are asked to interact with the application in the usability lab. All the usability data are collected by audio, video devices, testing automation tools and usability professionals. The following figure shows the layout of a typical fixed usability lab.

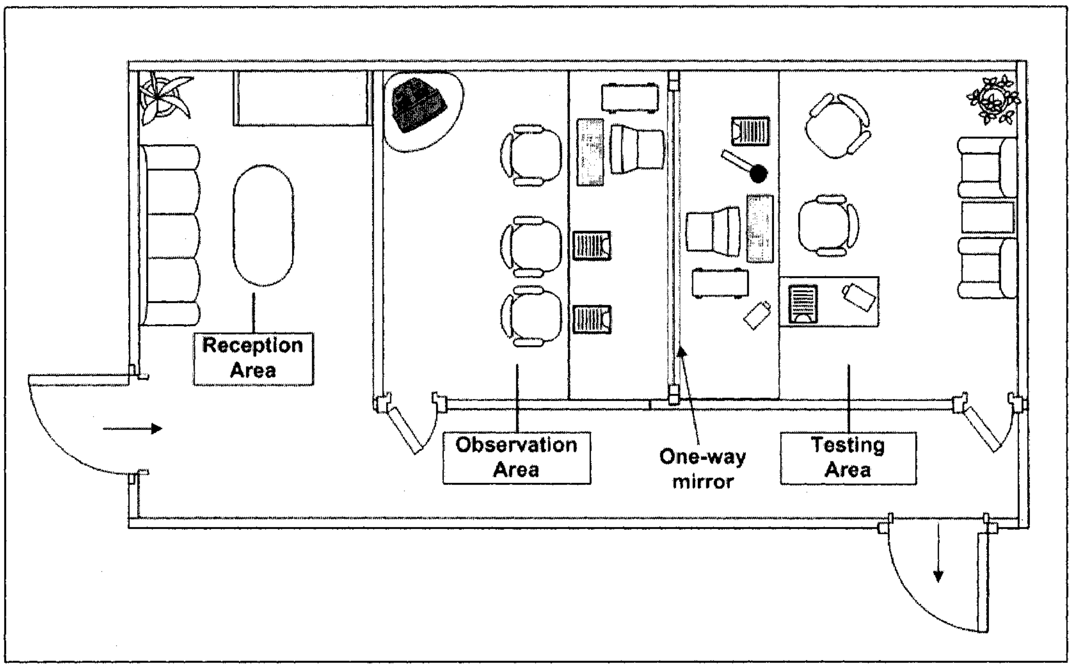


Figure 2-4: Setup of Fixed Usability Lab

The usability lab consists of three main components:

- Reception Area: the place where participants are welcomed. Participants also fill the pretest questionnaire, sign a Nondisclosure Agreement and a Tape Consent Form, and wait for test sessions in this area.
- Observation Area: a room, in which usability professionals, such as test technicians and observers, observe testing sessions through audio and video signals, captured by the microphone and cameras. Both participant's behaviors and screen that the participant interacts with can be observed. Observation area and testing area are separated by a one-way mirror and sound-proof walls, which allow the observers to see and study the participant's behavior by talking and exchanging ideas out loud without disturbing during the tests.
- Testing Area: a small room intended to simulate the participant working environment. In this room, the participant interacts with the system or application that needs to be tested. Audio and video devices, such as microphone and camcorder, are installed for collecting A/V signals during testing sessions. The usability evaluator sits beside the participant, conducting the usability testing

including distributing tasks and instructions, answering participant questions, and recording important usability events and data.

2.4.2. On-site Usability Testing

When carrying out on-site usability testing, usability professionals go to the users' actual working place to conduct usability tests with mobile usability testing equipment. Mobile usability testing equipment normally includes a laptop, a camera, and a camcorder. The laptop has been installed with both the application to be tested and all the software that are needed to conduct a usability test, for instance, a screen capturing application The camcorder and the camera are used to catch the user's behavior and verbal data. Test participants are asked to performance a set of predefined tasks in their real working environment by the instruction of usability specialists following a certain procedure. Figure 2-4 shows a portable usability lab for conducting on-site usability tests. [ALUCID Solution]

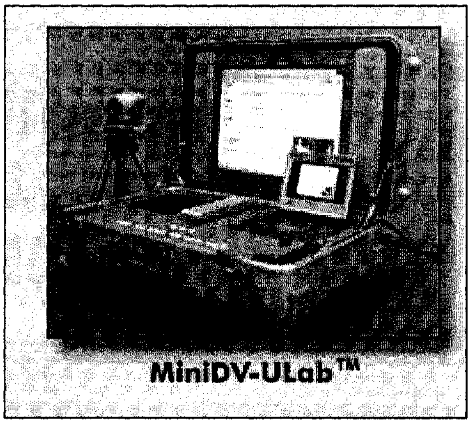


Figure 2-5: A Portable Usability Lab Proposed by ALUCID

2.4.3. Remote Usability Testing

Remote usability testing is relatively new when compared to the two others. Remote usability testing, also called remote evaluation is defined as a type of usability evaluation where evaluators are separated in space and/or time from users [Hartson et al. 1996]. Thanks to the world-wide Internet and network communication applications, usability professionals and test participants may not have to be together as the case in previous two kinds of testing. The following figure 2-5 shows the logic structure of conducting a remote usability test.

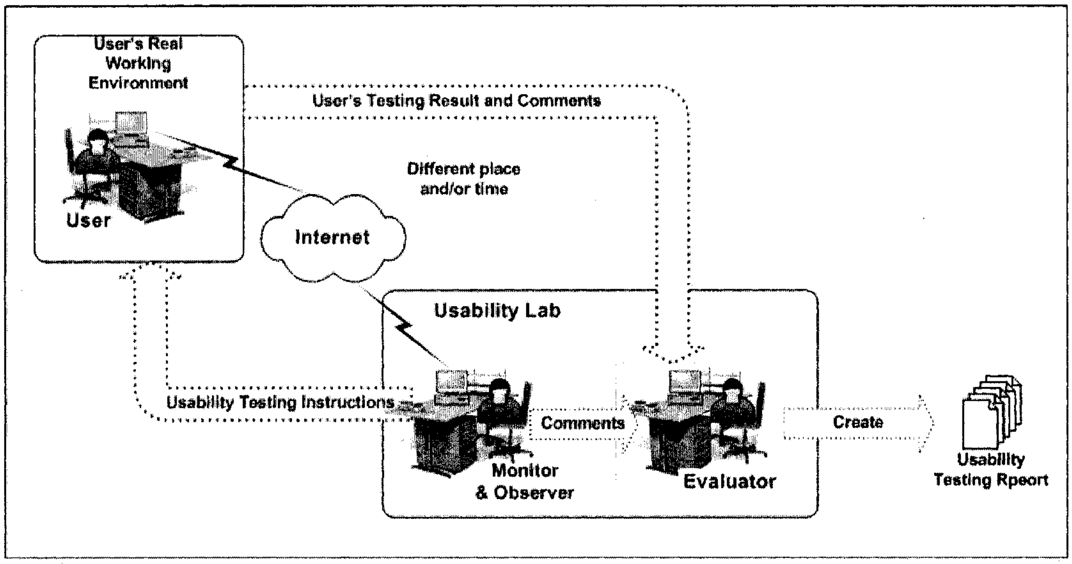


Figure 2-6: Structure of Remote Usability Testing

There are many ways to conduct remote usability tests that fall into two major categories, synchronous and asynchronous [Dray and Siegel 2004]. In the synchronous method, usability professionals interact with participants and usability data are received, observed and managed in real time. During the test, video conferencing and remote control applications are often used for data sharing, such as Microsoft NetMeeting. On the other hand, with asynchronous method, there is no interaction between the usability test conductor and participants during data collection. Usability professionals do not need to access data in real time.

2.4.4. Remote vs. Fixed and Mobile Labs

In last section, I introduced three different way to conduct usability tests. Now a question is raised: how can we know which kind of usability testing needs to be chosen. The answer depends mostly on the facilities we have. The facility includes the working space, equipment, and the software for analysis and observation. In this chapter I will compare these three types of usability tests in terms of testing facilities.

Limits of Fixed Usability Labs

To conduct local usability tests, one usually needs fixed usability labs, which are equipped with two rooms separated by a one-way mirror. In spite of usability lab's considerable contribution to the software engineering and the comprehension of the human computer interaction, one should also need to notice some weaknesses of the actual laboratories. The first constraint is that labs are costly in terms of needed equipments and software of analysis and observation. Only big companies or research centers can afford usability laboratories. The cost of the infrastructure is estimated to a range of 100,000$ US to 150,000$ US (management and test analysis software included in the estimation). Furthermore, the rooms have to be put at disposal.

The second constraint is that fixed labs force not only the participants but also the testers and the observers to gather in a same place and at the same moment. It appears difficult to recruit a representative sample of the target population in so far as the participants' transportation is costly and sometimes technically impossible. In addition, considering means put at disposal and the mobilized staff, the number of participants is limited because of essentially financial and availability issues. For these reasons, a test session generally does not count more than ten participants. The tests results are becoming less reliable from such a relatively little sample.

The third unavoidable constraint that one should indicate is that participants are taken away from their real daily working environment and put into a simulated environment, the usability lab, to test the application. Therefore, tests results can be biased. Test environment and the workstations are not familiar to the participants. The participants, surrounded by the observers and the testers, feel generally too anxious to achieve tasks which they might accomplish without being disturbed by others. This kind of things always happened during the usability tests in our lab. The environment and the emotional character of testers have a negative influence on the participant's behavior, and consequently reflect on the pertinence and the reliability of test results.

Mobile labs

A mobile usability lab comes generally in the form of a suitcase, which contains all necessary equipment for carrying out usability tests. In general, the suitcase contains a camera, a laptop, a camcorder and a video recorder. The equipment used in mobile labs is not excessively expensive. Rubin estimates the approximate cost of the basic equipment to be between 10,000$ US and 15,000$ US [Rubin 1994], which is only about one tenth of the costs of fixed labs. Moreover, this equipment is portable. When taking the test conducting software into account, the total cost goes between 15,000$ US and 20,000$ US. Moreover, the mobile usability lab is easy to install it in the participant's place and conduct the tests in the real participant's working environment.

Nevertheless, the observers and the testers still need to travel and the number of participants is still often limited due to the same reason mentioned in the case of fixed usability labs. In addition, even though the participants are in their own working places, they are still influenced by the presence of the observers and the testers.

After going over the advantages and the disadvantages of fixed and mobile labs, it is the time to compare them with remote usability tests.

Advantages of Remote Usability Tests

Remote usability testing is becoming more and more popular due to several obvious advantages over traditional lab testing [Moha et al. 2004b].

- It allows testing with numerous participants with diverse backgrounds. The tests are performed with reduced budget, and they take less time. In addition, the participants are almost not influenced by the test environment. During an empirical study carried out by Tullis, usability tests of websites have been performed within a traditional usability lab with 8 participants and with 29 remote participants via Internet with the aim of comparing and determining the efficiency of these two techniques [Tullis et al. 2002]. The study has shown that both techniques enable the testers to collect the most significant problems. However, remote tests produced the most reliable subjective affirmations of websites because of not only the greater number of involved participants but also the fact that users are not influenced by the test environment.
- Another advantage is that neither the participants nor the testers need to make costly and timely travel. Indeed, the testers test and interact with the participants

remotely. The testers themselves can also be geographically remote to each other and collaborate all together during the same test session. Consequently, it is possible to take advantage of the expertise of professionals and specialists, who are located all over the world. In addition, participants' sample can be broad, diverse and international, which is a determining advantage, especially if it is for developing specific applications that target a foreign population [Dray and Siegel 2004].

- Thirdly, since there is almost no environmental influence while conducting remote usability tests, participants feel more comfortable, and as a result, their comments are more reliable. Participants can remain anonymous, which makes their comments even richer. In addition, tests made in the participant's real work environment in his/her office or at his/her home reveal relevant problems related to specific work environments that could not be detected otherwise [Tullis et al. 2002].

2.4.5. Challenges of Remote Usability Testing

In spite of all advantages of remote usability testing we have presented above, there still are some challenges we need to take into concern. [Moha et al. 2004a]

It is difficult to control the participant's environment remotely. Indeed, participants can be distracted by either their family or their colleagues during the tests or suddenly stop the test session [Bartek and Cheatham 2003]. Nevertheless, if we look this from another perspective, this disruptive environment is part of the final user's environment. Thus, this loss of control can be considered as an advantage because the collected data truly reflect to the participant's work environment. When conducting remote usability tests, especially international ones, we must be prepared for the unexpected, such as system compatibility and stability, system crash, and network failure. When these things happen, we should be able to solve them remotely [Dray and Siegel 2004]. Sometimes, unfortunately, there is the risk that the problems cannot be overcome online.

In addition, the limited scope of visual feedback leads to information loss because of non verbal communication and participants' attitude (body language, tone of vice, and nonverbal sounders) [Dray and Siegel 2004]. Although the participant is monitored by a camera, only his/her face can be captured; the rest of the body in particular the hands and the room where the participant is located are hard to visualize.

Remote data transmission raises security and confidentiality issues. In some cases, participants' workstations can be located behind firewalls which are sometimes impossible to cross [Bartek and Cheatham 2003]. The firewalls have to be disabled or created a specific link in order to enable the access and the information sharing between participants and testers. Building trust with remote usability testing is a real challenge we are going to face.

In addition to the security issues, there are also performance ones [Bartek and Cheatham 2003]. In the case when participants remotely share an application with testers, thanks to some video conferencing tools such as IBM Lotus Sametime and Microsoft NetMeeting, the interaction can be slow because of network instability and a poor Internet connection. The situation will be even worse since would be many A/V signals need to be transmitted in real time.

The challenges addressed above can be partly solved by using some tools and increasing network speed and stability. Based on the comparison between usability labs and remote usability tests respectively, the latter ones, despite some limits proposed and under some realization conditions, suit well for certain types of tests. For example, remote usability testing are particularly adapted for Web applications, desktop and websites testing [Tullis et al. 2002] for the duration lower than three hours [Bartek and Cheatham 2003].

2.5. Current Usability Testing Processes

A great amount of existing methods help us assess usability of software applications and identify usability problems. However, the most efficient way is usability testing, which is typically carried out. The quality of conducting the usability tests has a major impact on the relevance of the collected data and the problems identified at the end of the usability tests rather than the equipments used to conduct those tests. When usability testes are conducted, a formal process is required to guide usability testers to accomplish their tasks. General speaking, a usability testing process is composed of a series of steps, which contain a set of activities. Several processes have been proposed and among the most cited in the literature are the Nielsen usability evaluation process [Nielsen 1994b], the Rubin testing process [Rubin 1994], and the Mayhew's Usability Engineering Lifecycle Process. Furthermore, the C&C usability testing process is a logical extension of Rubin's process. Detail discussions for these processes will be addressed in the following sections.

2.5.1. Nielsen's Usability Evaluation Process

The usability evaluation process model is not explicitly defined by Nielsen [1994b]. He regards usability testing as one step in his usability engineering lifecycle. (Details are shown in table 4.) In his usability engineering lifecycle model, Nielsen integrates usability into user centered design (UCD), a user interface design process that focuses on usability goals, user characteristics, and tasks in the design of an interface. But it does not emphasize the process of usability testing itself.

Nielsen Usability Process Steps	
1. Know the user a. Individual user characteristics b. The user's current and desire tasks c. Functional analysis d. The evolution of the user and the job 2. Competitive analysis 3. Setting usability goals a. Financial impact analysis	4. Parallel design 5. Participatory design 6. Coordinated design of the total interface 7. Apply guidelines and heuristics analysis 8. Prototyping 9. Empirical testing 10. Iterative design a. Capture design rationale 11. Collect feedback from field use

Table 2-3: Nielsen's Usability Engineering Lifecycle Model

Within the usability testing step, some important issues, which Nielsen considered, are dressed, such as test goals, test plan, and selecting representative users. However, the sequence of these activities is not discussed. Nielsen rather focuses on the conductstep of usability testing, consisting of four sub-stages: preparation, introduction, performing the test, and debriefing. Several methods or testing protocols are also introduced, for example, thinking aloud and constructive interaction. More or less, Nielsen's usability evaluation process is a guideline to lead one through some important issues to be considered when conducting usability testing.

2.5.2. Rubin's Usability Testing Process

Rubin [1994] presents his usability testing process taking the stand of a usability tester. The process focuses on activities, divided into six stages (Figure 2-6) that testers should perform when they conduct a usability test.

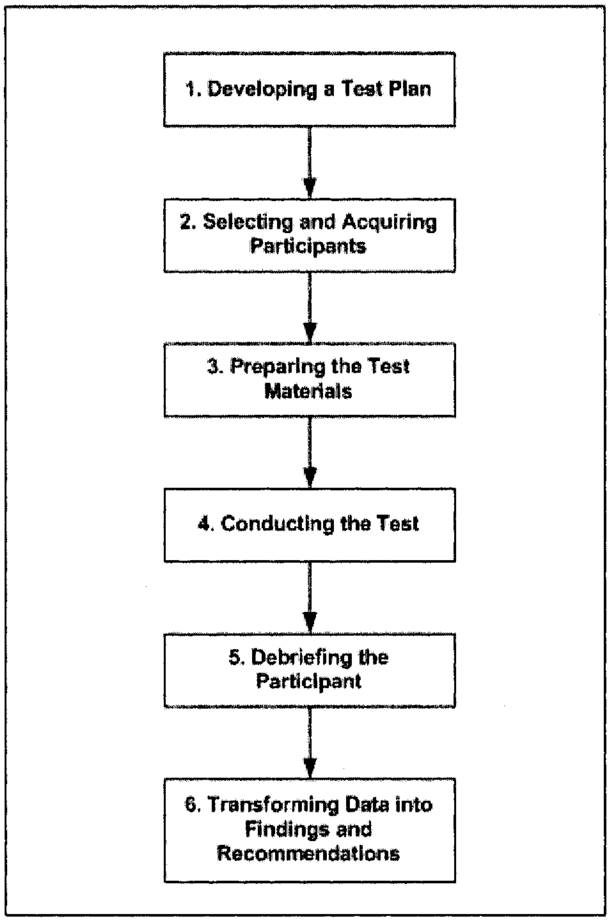


Figure 2-7: Rubin's Usability Testing Process

In the representation of Rubin's process, although he covered some important milestones of usability testing, the test actors and their roles are not explicitly emphasized and the following steps are somehow missing or implicit: design, pilot test, compile and analyze. The process does not specify the step of designing the tests that defines how the tests will be performed. Moreover, before conducting the real tests, it is necessary to conduct some pilot tests or pre-tests in order to ensure the smooth functioning of the real tests. Finally, the distinction between transforming data into findings and recommendations, and the analysis is not differenced at all. These steps should not be implicit; they are valuable milestones that need to appear in the process model.

2.5.3. C&C Usability Testing Process

Computing & Communications UCD group at University of Washington (C&C) proposes a 4-steps process that includes a detailed description of actors involved and their roles. This process is summarized in the following table.

Process Step	Actors Involved
Step 1. Plan the test	Usability Engineer, Designer/Developer, Usability Coordinator
Step 2. Design the test	Usability Engineer, Designer/Developer, Usability Coordinator
Step 3. Conduct the test	Usability Engineer, Designer/Developer, Usability Coordinator
Step 4. Analyze and report the test results	Usability Engineer, Designer/Developer

Table 2-4: C&C Usability Testing Process

The definitions of actors are:

- **Usability Engineer:** A person having formal training in usability testing who is responsible for evaluating the product and managing the test;
- **Designer/Developer:** One or more people responsible for designing the product being evaluated;
- **Usability Coordinator:** A person responsible for recruiting and scheduling participants for the test.

C&C usability testing process explicitly states the actors and their roles in each step of the testing process, but lacks for the methods and techniques used at each stage to collect the data or perform the activities. Furthermore, the list of actors is not complete, for instance, representative users who were selected from the target audience for participating the test is missing. Finally, the three defined actor categories, Usability Engineer, Usability Coordinator, and Designer/Developer are too general, and need to be decomposed and made them more specific.

2.5.4. Mayhew's Usability Engineering Lifecycle Process

Another well-known process model is Mayhew's usability engineering lifecycle model. Mayhew integrates usability tasks with an overall product development methodology on a project-by-project basis [Mayhew 1999]. Her model represents an engineering approach to achieve usability during product design and development which includes:

- Structured usability requirements analysis tasks,
- An explicit usability goal setting task, driven directly by requirements analysis data,
- Tasks supporting a structured, top-down approach to user interface design that is driven directly from usability goals and other requirements data,
- Objective usability evaluation tasks for iterating towards usability goals.

Figure 2-7 shows the process diagram of the proposed usability engineering lifecycle [Mayhew 1999]. It indicates all lifecycle tasks, and approximately where each one should be applied within either a modern rapid prototyping or an object oriented software engineering (OOSE) methodology. It also shows how Usability Engineering Lifecycle tasks occur in parallel and tightly intertwined with traditional development tasks.

Within the evaluation steps, Mayhew provides three general stages for each test iteration:

- Plan the test and develop supporting materials.
- Run the test users and collect data as specified in the test plan.
- Analyze and interpret the data and formulate redesign recommendations.

In every stage, she also gives some activities that need to be done. For instance, in the planning and preparing stage, nine activities are addressed including decision on ease of learning/ease of use focus for the rest, decision on user and task focus for the test, design test tasks etc.

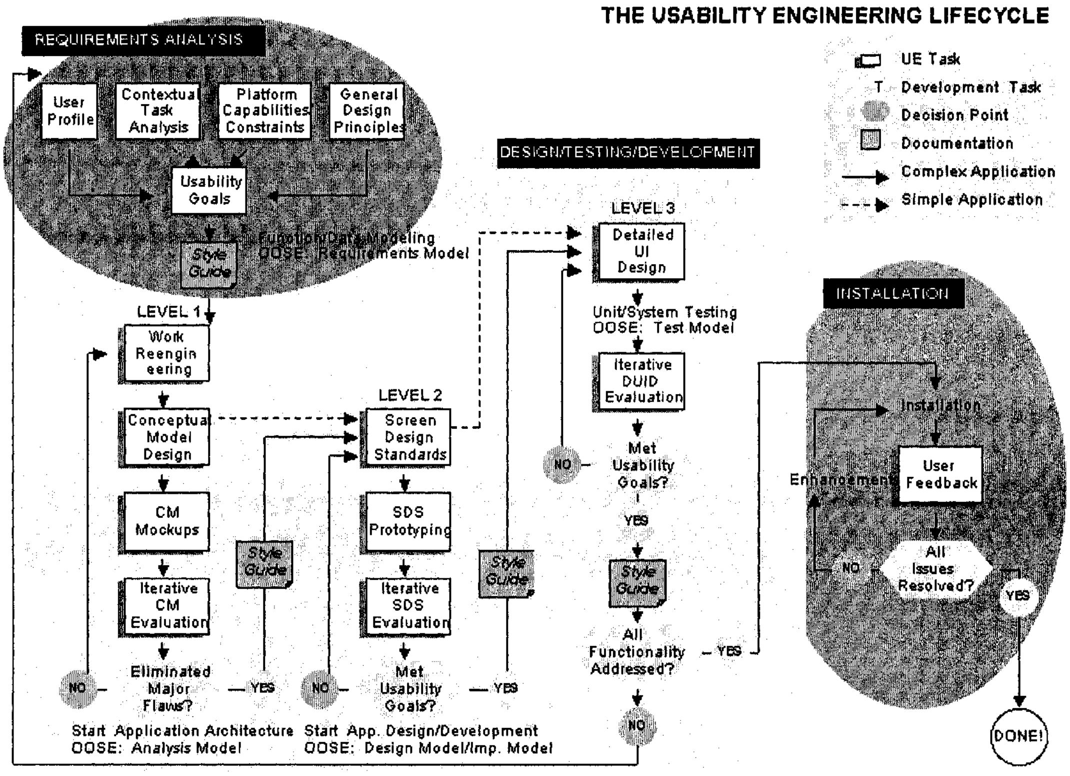


Figure 2-8: Mayhew's Usability Engineering Lifecycle Model

Although Mayhew's process model is not focused on usability testing, and most of the time she mixes usability evaluation with iterative design and development activities, this model still has many advantages and good concepts that we can borrow. The main advantages of this model are:

- Provides techniques for carrying out each lifecycle task.
- Offers samples and templates for accomplishing tasks.
- Defines the roles, which are usability engineer, user interface designer, user interface developer, and user, and resources that related to a specific lifecycle step.

- Provides level of effort that gives a rough idea of effort estimates.
- Raises a good concept: tailor the process to a particular project. Since there are many difference from one project to another, tailoring or adapting of the process is really important. It can let the process fit the project and reduce the workload.

Besides the advantages mentioned above, however, there are also limitations in terms of usability testing.

- A gap between process and usability tools that can be used to automate some tasks.
- She does not combine roles with activities within a usability testing step. Moreover, the classification of roles is two general. For example, the role of usability engineer can be divided into some sub-categories, such as evaluator and data analyst.
- The description of the different steps of the model is too general: the sequence of the activities is missing. Some of the activities can be done parallel, for instance, recruiting test users and designing and assembling test environment.

2.6. Further Discussion about Weaknesses of Current Processes

With a decade of usability practices in industry and academic research, many methods, techniques, and automation tools have been proposed and developed. Many documents have been published providing experience and lessons. A detailed discussion about usability methods and tools will be given in the next chapter combined with newly proposed usability testing process model. Tools and methods release usability professionals from tedious and time-consuming work, such as data collecting and analyzing. That makes it possible for them to pay more attention on usability testing itself.

Although usability specialists have more confidence in the testing process than they did in the early years, usability evaluation processes are still informal. The confidence has grown because the data collection and analysis methods have been validated over times but the need of productivity (doing more with fewer resources) and faster development and release cycles tend to lead to less formality in the process [Dumas and Redish 1999].

To some extent, the existing HCI/usability testing processes are incomplete and informal. Most of them are just the summary of working experiences gathered while conducting the usability tests. Three major problems still exist in current testing processes. The first one is that some major milestones were not well defined in the current usability testing processes [Mayhew 1999;Nielsen 1994b;Rubin 1994]. These milestones will be discussed in next chapter include hiring participants, compiling the data results collected from different sources as well as conducting pre-tests. One needs to pay more attention on these milestones as they may have a major impact on the accuracy of the tests results.

Secondly, these processes do not provide answers to the key question in process modeling: *who is doing what, when, and how?* Actors, who are involved in usability testing, are not well defined or even not defined at all. The roles that each actor performs in a given step are not clearly indicated.

The third problem is that there is a big gap between usability testing processes and methods and tools that can be applied in usability testing. Even through, some processes compose some usability methods, the problem is that they only provide general description and do not cohere with a specific step. When others want to apply these processes, they still do not know how and when to use the methods.

In the next chapter, I am going to propose a new usability testing process model that aims to solve these three major problems. Activities, tools/methods, outputs and actors are four basic building blocks of the process model.

3. The Proposed Usability Testing Process Model

3.1. Introduction

In the previous chapter, I gave a literature review of current usability testing processes, and also discussed the three major existing problems related to these processes.

1. Some milestones are missing in current usability testing processes.
2. Current processes do not provide answers to the key question in process modeling: who is doing what, when, and how?
3. There is a big gap between usability testing processes, methods and tools that can be applied in usability testing.

In order to solve these problems, we proposed a new usability testing process. This proposed process not only combines strengths from existing processes, but also presents a new presentation structure of process. The process model is defined based on the four sources of information: existing usability testing process, literature review on usability testing, tips/hints from usability experts, and HCI standards (Figure 3-1). This process is defined for general purpose usage. It can be customized for special using, such as remote usability testing, low fidelity testing (e.g. walkthrough with a paper-based prototype). Some of steps can be removed according to the test context.

In a well-defined process, we should specify the *activities* required for each step of the process, the *actor* (human or system) involved in each step, the *methods* to be used, and *tools* to perform the activities as well as the *artifacts* produced and used in each activity. Activities, actors, tools, methods and artifacts are the basic ingredients to define and formalize a process. The processes presented in the previous chapter fail to detail these elements to some extent.

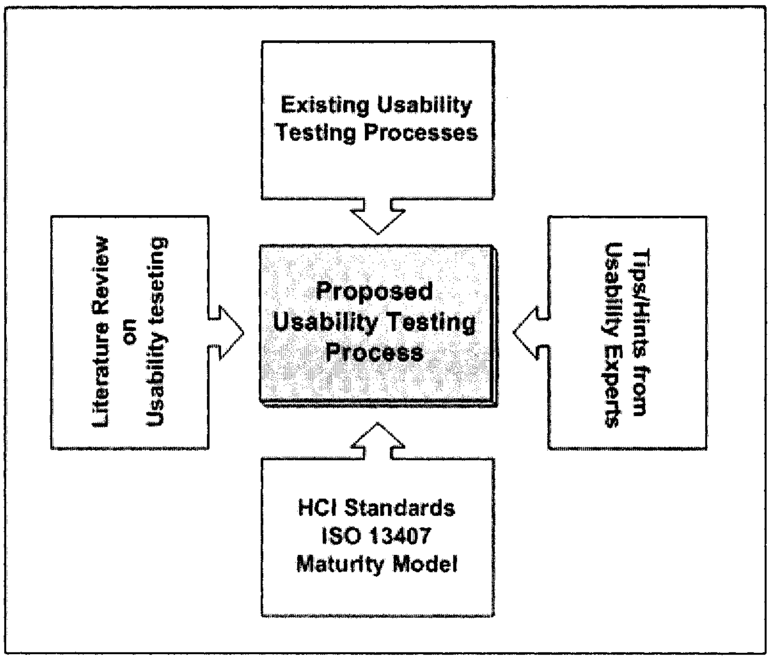


Figure 3-1: Usability Testing Modeling

Based on the concerns addressed above, we defined four basic elements in our usability testing process model to answer the question: *who is doing what, when, and how.*

- ***Actors, the 'who':*** an actor defines the behavior and responsibilities of an individual, or a group of individuals working together as a team on an activity or a set of activities.

- ***Activities, the 'what':*** an activity of a specific actor is a unit of work an individual in that role may be asked to perform. For each activity, it has specific inputs, and after performing this activity, some artifacts or outputs are generated. An artifact is a piece of information that is produced, modified, or used by an activity.

- ***Process Workflow, the 'when':*** every step and activity has positions in the process workflow, which shows the sequence of performing. By following the workflow, usability engineers know exactly the time to do their tasks, parallel or sequentially.

- ***Methods and Tools, the 'how':*** which give the way for usability engineers to accomplish their work. Tools help to automate some activities, such as data collecting, and transform inputs into artifacts during an activity or a step.

	1. Plan	2. Design			3. Acquire	4. Setup
Main activities	1.1. Define Test Purpose and Goals 1.2. Define Target Audience 1.3. Define Test Schedule and Deliverables 1.4. Define the Business Plan	2.1.1. Refine Target User Profile and Characteristics 2.1.2. Divide User Profile into distinct Users Categories 2.1.3. Define the Number of Participants for each category	2.2.1. Define Target Product and Features to be Tested 2.2.2. Select and Adapt Test Methods 2.2.3. Define Types of Data and Metrics 2.2.4. Select and Adapt the Data Collection Methods 2.2.5. Select and Adapt the Measurement and Data Analysis Techniques 2.2.6. Define the Usability Test Team and Testers Profiles	2.3.1 Define Testing Environment 2.3.2 Define Equipment and Software 2.3.3 Prepare the product or the prototype 2.3.4 Define Task Scenarios 2.3.5 Prepare Prerequisite Training Materials 2.3.6 Prepare Test Materials 2.3.7 Detail Test Procedure	3.1 Contact and Interview Participants 3.2 Build the List of Participants 3.3 Define the Test Sessions Schedule	4.1 Install and Review the Product to be Tested 4.2 Prepare the Test Physical Environment 4.3 Check the Equipment 4.4 Review and Assemble the Test Materials 4.5 Check the Availability of Participants
Artifacts	- Purpose and Usability Goals - Target Audience Profile - Test Schedule - List of Deliverables - Required resources and budget - Testing Plan	- End User Persona - Matrix of Users Categories - List of participants	- Specification of the Product or the Prototype to be tested - List of Methods - List of Data Types and Metrics to be collected - Data Collection Methodology - Data Analysis Methodology - Usability Coordinator, Usability Evaluator and Observer Profiles	- Test Environment Description - Test Equipment Description - Product or Prototype with documentation - Tasks Scenarios - Training Materials - Test Procedure Test Materials: - Screening Questionnaire - Test Process Overview Script - Background Questionnaire - Data Collection Instruments and Forms - Nondisclosure Agreement and Tape Consent Form - Post-test Questionnaire - Debriefing Topics Guide	- List of Participants - Test Sessions Schedule	- Check List of Test Environment - Check List of the Status of Equipment - Revised Version of Test Materials - Final Test Session Schedule
Methods	- Surveys - Interviews - Old Design Analysis - Competitor Analysis - Stakeholder Meeting - Participatory design - Analyse Context of Use - Requirements Meeting - Usability Planning	- Surveys - Questionnaires - Interviews - Participatory Design - Task and User Analysis - Contextual Inquiry	- Guidelines on Selecting Methods	- Check List of Materials - Development Guidelines for Task Scenarios - Prototyping - Task Analysis	- Internal Resources (participants databases) - Advertisements - Employment Agencies	- Check Lists - Contact by email or phone
Actors	Manager / Coordinator	Manager / Coordinator Evaluator	Manager / Coordinator Evaluator Monitor	Manager / Coordinator Evaluator Monitor	Recruiter User / Participant	Manager / Coordinator Monitor

Table 3-1: Usability Testing Process Table - 1

	5. Preview	6. Conduct	7. Debrief	8. Compile	9. Analyze	10. Report
Main activities	5.1 Conduct a Pre-test Yourself 5.2 Conduct a Pilot Test 5.3 Conduct a Formal Inspection 5.4 Provide training to usability testers and observers	6.1 Initialize the Environment and the Equipment 6.2 Welcome the Participant 6.3 Ask Participant to Fill out and Sign any Preliminary Documents 6.4 Explain the Test process overview 6.5 Provide any Prerequisite Training 6.6 Ask the Participant to Fill out Any Pre-test Questionnaires 6.7 Distribute or Explain the Task Scenario(s) 6.8 Monitor the Test Session and Record Data 6.9 Ask the Participant to Complete all Post-test Questionnaires	7.1. Collect Informal Open-mind Opinions from the Participant 7.1 Review the Post-test Questionnaires with the Participant 7.2 Interview the Participant 7.3 Thank the Participant 7.4 Review the Post-test Questionnaires alone 7.5 Add the Usability Testers and Observers Comments 7.6 Organize and Aggregate the Test Results	8.1 Cleanup and Organize the Raw Test Results 8.2 Annotate and Store Data and Test Benchmarks	9.1 Synthesize and Visualize the Tests Results 9.2 Identify and Prioritize Preliminary User Problems and Findings 9.3 Conduct Comparative Analysis 9.4 Identify Patterns and Findings 9.5 Summarize the all Analysis Results	10.1 Transform the Findings and Patterns into Recommendations 10.2 Prepare Preliminary Report 10.3 Select Video Demonstrations Clips 10.4 Develop and Prioritize the Recommendations 10.5 Produce Final Report 10.6 Produce or Update Design Guidelines 10.7 Follow-up with Observers and Usability Testers
Artifacts	- Quantitative and Qualitative Data - Data Collection sheets	- All Filled Questionnaires and Preliminary Documents - Quantitative and Qualitative Data - Data Collection sheets - Videotapes - Screenshots - Test session record	- Users Open-mind Opinions - Usability Testers and Observers Comments grouped by product, by user, by category of users - Answer Sheets of the Interviews - Post-test Questionnaire filled - Videotapes of the Debriefing Sessions	- Data Files for Analysis - Measurements: task timings, number of times help was accessed, etc. - Backup Data for Long-Term Studies	- Patterns - Statistical Data - List of the problems by Priority	- Preliminary Report - Video Demonstrations Clips documenting significant Events - Final Report synthesizing the major weaknesses of the test and the recommended solutions - Design Guidelines
Methods	- Formal Inspection Meeting to review the Pilot Tests	More than 50 methods can be used among the following: - Videotape test sessions - Participatory Design - Wizard of Oz - Thinking Aloud Protocol - Heuristic Evaluation - Walkthroughs - Inspections - Card-Sorting - Ethnographic Study / Field Observation - Remote testing	- Guidelines on Questioning Participants - Methodology for Conducting a Debriefing Session - Videotape Debriefing Sessions - "Replay the Test" Technique - Reviewing Alternate Designs - "What did you remember?" Technique - "What would you do if?" Technique - "Devil's advocate" Technique - Usability and satisfaction survey	- Transfer handwritten notes to a computer - Transfer times and other quantitative data onto electronic sheets - Review videotapes and notes	- Use Descriptive Analysis Techniques to Synthesize and Visualize the Tests Results - Data Mining / Discovery / Prediction - Statistical data analysis techniques - Compare time on tasks, errors, satisfaction and any other measures - Comparisons against benchmark goals, between Product Versions, Design Alternatives, Competing Products and Group of Users - Prioritize problems by criticality: categorize a problem by severity, rank the problem by estimated frequency of occurrence, and then ascertain a problem's criticality.	- Guidelines on Developing Recommendations - Guidelines on Producing the Final Report
Actors	Manager or Coordinator Evaluator Monitor Data Analyst Users / Participants	Manager or Coordinator Evaluator Monitor Data Analyst Observers Users / Participants	Evaluator Observers Users / Participants	Evaluator Data Analyst	Evaluator Data Analyst	Manager or Coordinator Evaluator Observers

Table 3-2: Usability Testing Process Table - 2

Table 3-1 and 3-2 demonstrate the proposed usability testing process. Each column of the tables refers to a step or a sub-step of the process. Our proposed usability testing process is divided into ten steps, from *plan* to *report*. The design step contains three sub-steps. The numbers in front of the step names imply the sequence of steps except acquire and setup steps, which can be done simultaneously and are demonstrated in the table 3-1. Detailed information about step conducting sequence is presented in process workflow section.

The first row of the two tables denotes main activities that need to be performed in a particular step. The leading numbers show performing sequence of the activities. The second row refers to all artifacts related to a specific step. Artifacts include the following components:

- Formal documents generated from a step;
- Usability templates;
- Outputs of an activity; and
- Input of an activity from other steps or activities.

The third row refers to the methods that can be applied in a step and the fourth row denotes the actors who involve in a step.

The following four sections detail the proposed usability testing process model in the sequence of steps and activities, usability methods and tools, actors' roles, and process workflow.

3.2. Steps and Activities

Based on the analysis of all the existing processes listed in the previous section, we propose a 10-steps process model. Compared to other processes that propose five or six steps, we add four extra steps that are missing or not explicitly defined in the processes listed previously. These new steps are: (1) Designing the tests; (2) Making a preview of the tests; (3) Compiling the results of the tests; and (4) Analyzing the results of the tests. We argue that by making them as milestones and giving them the same importance as other steps, we can improve the conduct of the tests. The major steps of the proposed model are briefly described here.

1. ***Plan.*** This activity consists of producing a testing plan which answers the following questions: what, why, how, when and where to test;
2. ***Design.*** This step is composed of three sub-steps. The first one is related to determine what kinds of participants are required to attend the test. The second sub-step selecting and adapting the research methods, and the third one is to define testing environment and prepare testing materials;
3. ***Acquire.*** Generally needed for large tests, this step consists of selecting, hiring and interviewing participants.
4. ***Setup.*** During this step, the hardware equipment and the software tools needed for tests are deployed, installed, configured, and tested.
5. ***Preview.*** Most often neglected even if highly recommended by experts, this step consists of conducting a series of pilot tests to ensure that the test environment, materials, and resources are appropriate and functional. It is also an opportunity for any last time refinement of the test design and the acquisition of additional equipments and materials.
6. ***Conduct.*** The real test is performed in this step where several quantitative and qualitative data are gathered including participant feedback, video observations, and screen snapshots.
7. ***Debrief.*** This step refers to interviewing the participants, reviewing with them their reactions during the test, and getting their feedback.
8. ***Compile.*** The data are aggregated, consolidated, annotated, and properly archived to facilitate their later retrieval and analysis.
9. ***Analyze.*** Using appropriate data analysis and data mining techniques, this step aims to transform the qualitative and quantitative test results into findings and patterns.
10. ***Report.*** This step consists of transforming findings and patterns into recommendations. This step ends up with a report that states all the findings and recommendations.

Furthermore, we define the *artifacts* produced and used in each step and activity. For example, the plan step produces a test plan as an output. For the conduct step, we list the following artifacts: all filled questionnaires, data collection sheets, videotapes, screenshots, etc. It is important to well-structure the user-oriented testing process in order to ensure that it is going to be well-understood and applied.

3.3. Usability Methods and Tools

3.3.1. Methods

The term method here is used to refer to a mean or manner of procedure or techniques, especially a regular and systematic way of carrying out one or more activities in different stages to achieve a specific goal. There can be methods for conducting usability tests, interviewing the participants, as well as specific methods for measuring the user performance and satisfaction.

Over the last 15 years, a large set of methods has been proposed. Ivory and Hearst [Ivory and Hearst 2001] analyzed and classified the most popular methods from the automation perspective. They classified usability evaluation methods into 5 categories as follows.

- ***Testing***: an evaluator observes users interacting with an interface (i.e., completing asks) to determine usability problems.
- ***Inspection***: an evaluator uses a set of criteria or heuristics to identify potential usability problems in an interface.
- ***Inquiry***: users provide feedback on an interface via interviews, surveys, and the like.
- ***Analytical Modeling***: an evaluator employs user and interface models to generate usability predictions.
- ***Simulation***: an evaluator employs user and interface models to mimic a user interacting with an interface and report the results of this interaction (e.g., simulated activities, errors, and other quantitative measures).

Table 3-3 gives an example of the methods defined for the two first steps of the process. For each step, we proposed a collection of methods to use in order to successfully perform the step. The methods selected represent established approaches, which have been successfully applied within the industry. Among these methods, the ones that we have been using in some of our empirical studies are card sorting, cognitive walkthroughs, and persona [Naghsin et al. 2003].

Plan	Design
List of methods: • Surveys • Interviews • Old Design Analysis • Competitor Analysis • Stakeholder Meeting • Analysis Context of Use	List of methods: • Surveys • Interviews • Participatory Design Task and User Analysis • Contextual Inquiry

Table 3-3: Methods for the Plan and Design Steps

Bevan [2003] gives the detailed summary of mostly used usability methods and their usages.

3.3.2. Tools

The term tool means software applications that can assist usability testers to automate some tasks, such as data collection and analysis, which reduce workloads.

Companies and usability groups tend to develop a small assortment of tools support for their own, in-house use usability studies; meanwhile, most other tools currently used are not dedicated for usability tests. These tools only suit for one or several usability testing steps. Moreover, we are lack of documents or literature reviews of usability tools to give a state of art. A big gap has thus appeared. On one hand, many good tools exist that can be used in usability tests and can reduce workloads tremendously. On the other hand, usability professionals are fighting their way to conduct tests without fully tool support since they are not well-informed or are not familiar with tools selecting.

Under this situation, a wide range of tools survey is conducted within our group. We consult usability experts via personal mails, search on the web and compare technical specifications of new era's state-of-art tools and technologies, attend some demo sessions of the tools organized by the vendors and suppliers, and finally select the necessary tools fitting usability testing process.

We classify usability tools into 10 categories as follows.

1. ***Testing Process Management and Improvement.*** Manage usability testing process, show the testing status, and also store test object information.
2. ***Participant Recruiting.*** Help to create participants database, provide usability characteristics of participants that makes it easy to select suitable participants for specific requirement of usability tests.
3. ***Usability Measurement.*** Automatically evaluate the software application or websites after collecting required usability data. For example, SUMI measurement tool is one of most widely used tools. It gives usability assessment results by analyzing SUMI questionnaire feedback.
4. ***Remote Control.*** Give usability testers the opportunities to access test participants' computer remotely, which can be used to install software, observe participant's screen, and troubleshooting during remote usability testing.
5. ***Video/Audio Capture, Indexing, Storage and Retrieval.*** Fully or partly the whole A/V processing procedure, which includes capturing, storing, indexing, annotating and retrieving steps.
6. ***Video-Conferencing.*** Can be used for conducting or observing remote usability testing.
7. ***Online Surveys and Questionnaires.*** Assist usability professionals in creating, conducting, distributing and collecting their surveys and questionnaires online.
8. ***Automated Usage Capture.*** Automatically or semi-automatically collect usability data during the test session, such as the time that the participant accomplishes a task, or the access trace of visiting a web site.

9. ***Prototyping and Simulation.*** Generate prototypes or simulate the application that needs to be tested.

10. ***Data Analysis and Mining.*** Analyze the usability data that was collected during usability tests, to find useful patterns findings, based on which evaluator can generate the recommendation report.

Table 3-2 gives examples of tools for each category. After conducting the survey of usability tools, we select the most representative tools of each category, and make a fully investigation for them. Some tools such as OvoStudio and Noldus Observer cover a big range of the test process while others cover only one step. A detailed description of advantages and disadvantages of each of these tools can be found at Appendix A: Tools Survey. Figure 3-2 gives the usage range of tool examples of the ten categories in usability testing process.

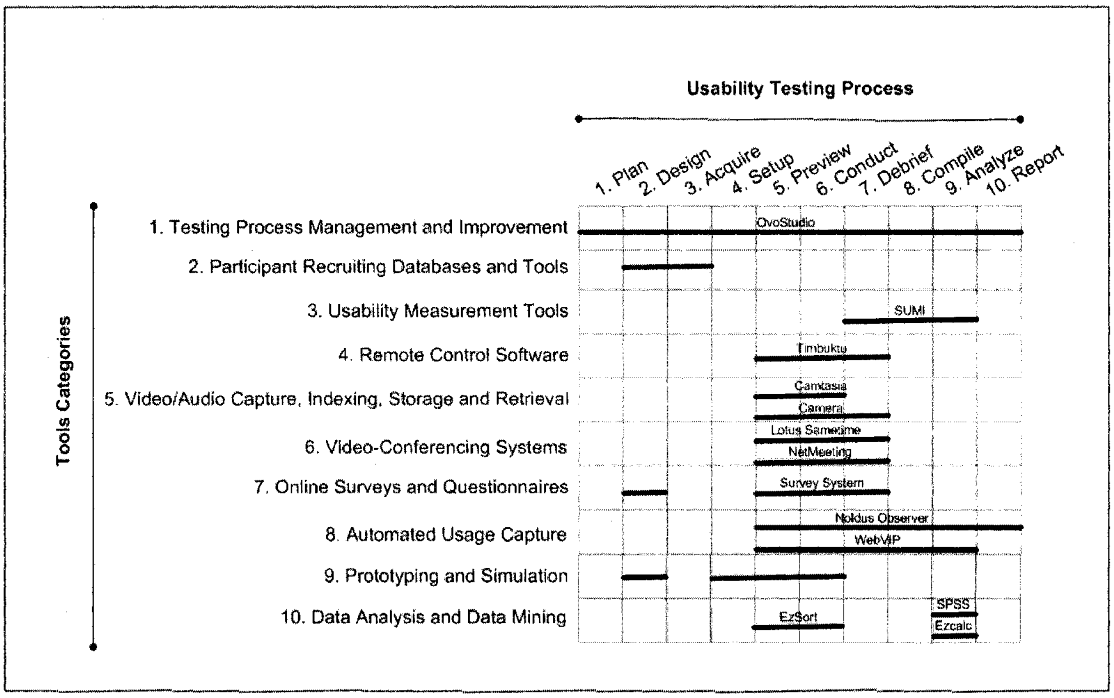


Figure 3-2: Example of Tools Usage Range

Categories of Tools	Tools examples
Testing Process Management and Improvement	▪ OvoLogger
Participant Recruiting Databases and Tools	▪ PopRecruit
Usability Measurement Tools	▪ SUMI ▪ MUMMS
Remote Control	▪ Timbuktu ▪ PCAnywhere
Video/Audio Capture, Indexing, Storage and Retrieval	▪ MPEG-7 Standard, ▪ IBM VideoAnnEx Annotation Tool, ▪ Adobe Premiere
Video-Conferencing Systems	▪ Microsoft NetMeeting ▪ IBM Sametime
Online Surveys and Questionnaires	▪ Ezpolls, ▪ QUIS, ▪ Question Master,
Automated Usage Capture	▪ Noldus Observer ▪ OvoStudios ▪ WebVIP & FLUD ▪ Camtasia Studio.
Prototyping and Simulation	▪ IBM EasyChart, ▪ MS PowerPoint, ▪ PatchWork
Data Analysis and Mining	▪ SPSS tools, ▪ IBM EZSort, ▪ WebCAT, ▪ SAS Data Mining Solution

Table 3-4: Examples of Usability Tools

At the end of 2004, after we finished conducting our usability tools survey, a great breakthrough happened in the usability testing field that TechSmith Corporation launched a powerful tool for conducting and analyzing steps of usability testing, Morae.

Morae can be defined as a semi-remote usability testing tool. It provides a software-based solution for usability testing of software, web sites, intranets and eBusiness applications. By using Morae, usability specialists can not only record the participant's computer screen and their behavior video signals, but also are enabled to create automatically a synchronized index of events occurring behind the scenes in applications and in the operating system [TechSmith 2004]. Usability specialists also can add markers, which highlight the participant's interactions or behaviors, during the test or at the data analysis stage, and these markers are automatically combined and synchronized with the audio/video signal in the real time. Markers make data analysis stage easier for data analysts when they are trying to find usability patterns.

Morae is composed of three components: Morae Recorder, Morae Remote Viewer, and Morae Manager. These three components are working together to provide fully support for conducting and analyzing stages of usability tests. Figure 3-3 demonstrates the physical localization of these three components.

When usability specialists use Morae in their usability testing, the whole process can be divided into three steps: conducting, analyzing and demonstrating. In the first step, the participant is interacting with the computer that has installed with Morae Recorder. The Morae recorder records screen, participant audio and video, and system events. These data are automatically synchronized by Morae Recorder, and streamed to observers' computers, where Morae Viewers are running. At the same time, observers are using Morae Viewer to create markers, comments or notes, which are sent back to the participant's computer and synchronized with other data. After test session finishes, all data will be stored in a file with .rdg postfix that can be opened by Morae Manager for data analysis. In analysis step, Morae Manager isolates the interesting patterns combined with the participant's A/V signal, and then exports them into AVI or WMV files. The last step is using Morae Player to demonstrate test findings. Figure 3-4 illustrates how Morae Recorder, Manager, and Remote Viewer work together.

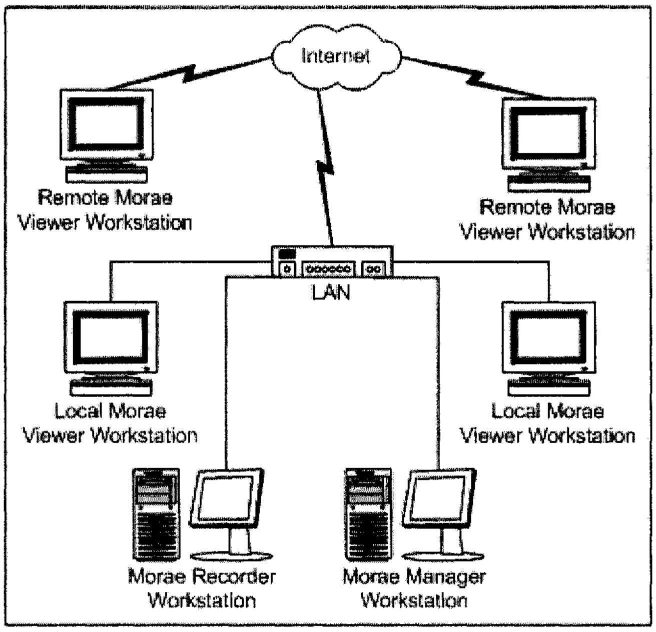


Figure 3-3: Physical Localization of Morae Components

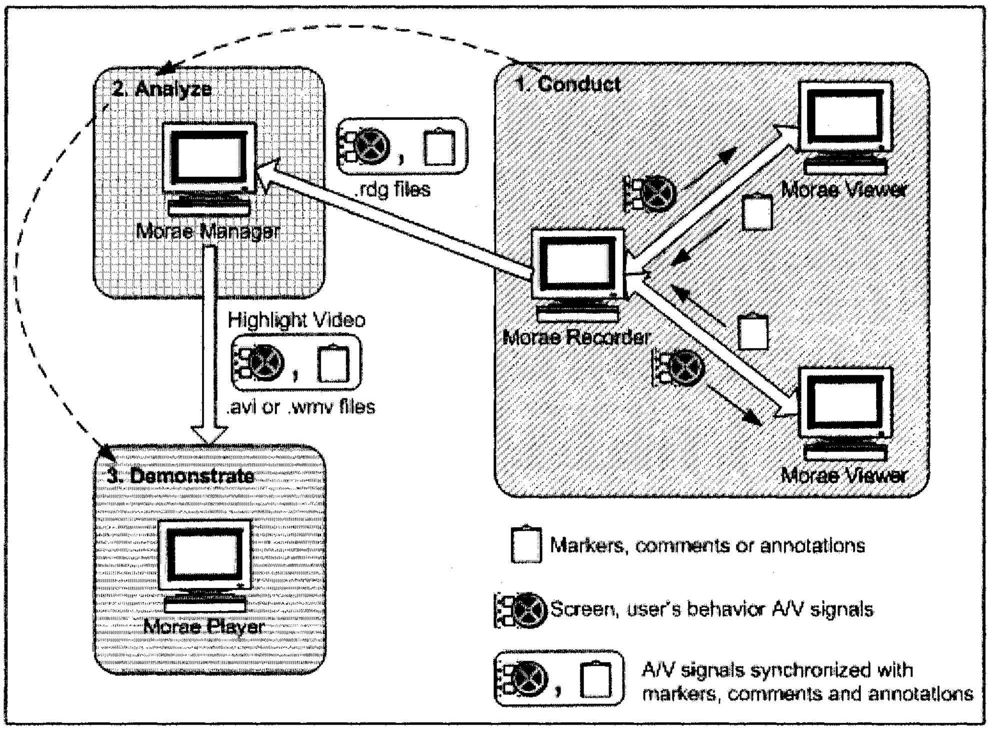


Figure 3-4: Morae Workflow

Besides a number of advantages, however, Morae still has two major drawbacks:

1. Morae does not fully support remote usability testing, which means Morae cannot be remotely accessed. Usability specialists need to install Morae Recorder on participant's computer. It just allows observers to monitor test sessions remotely.
2. Morae only covered two steps of usability testing process, which are conduct and analyze steps.

3.3.3. Templates

In addition, there is a special kind of tool, template. A template gives a document model for writing same kind of documents. It contains necessary information for the documents and gives hints on what kind of information should include and where to put it. For instance, Usability & User Experience Community of Software Technical Communication (STC) offers a series of templates for usability testing. The detail information can be seen through the following address: http://www.stcsig.org/usability/resources/toolkit/toolkit.html. National Institute of Standards and Technology (NIST) provides another very useful Common Industry Format for Usability Test Report (CIF, see Appendix B). Templates provide us the opportunities to reuse the knowledge, good experience and practices from others.

Usability methods and tools provide the answer to the *how* part of the question: *who is doing what, when, and how*.

3.4. Actors' Roles

One of the major weaknesses of the processes listed in the chapter 2 is that the roles of different actors involved in usability tests are not clearly defined. A role is defined as a set of specified responsibilities. An accurate definition of the roles of those actors can help to identify the expertise and skills required for achieving the tasks of the different steps in the usability testing process. For example, the coordinator who is responsible for hiring the participants needs to be familiar with the methods related to the selection of the participants such as the conception of questionnaires.

This section is going to provide the answer to the *who* part of the question: *who is doing what, when, and how*. The relationship between roles and people is not one to one relation, meaning that one person can take on more than one role in a usability testing project, and more than one person can share that role. Roles definitions of actors are listed as follows.

- ***Managers/coordinators:*** They are the usability team project leader. They are indeed the managers of the usability study as they assign roles to usability team members and coordinate their works.
- ***Monitors:*** They own the technical knowledge of the equipment and the software used in the tests. The monitors are responsible for installing and configuring the lab, and during the tests they have to ensure the smooth functioning of the test equipment.
- ***Recruiters:*** Their tasks are to select, recruit and schedule the participants for the tests. They can also set up the materials for conducting the tests with the monitor.
- ***Evaluators:*** They conduct testing, analyze and report the results. They are the primary persons responsible for testing the product design, analyzing, documenting the results and presenting the report to the development team.
- ***Data Analysts:*** They attend preview, conduct and debrief steps. The Data Analysts also help Evaluator to compile usability data into electronic format. During the test session, they use predefined data collection sheets or computers to record all usability data needed to be collected. They also help the evaluator to record important events when participants interact with the system.
- ***Observers:*** They observe the users during tests sessions; they do not intervene during the tests (usually they sit behind a one way mirror or watch the test through a video signal), but provide their feedback at the end of the session. Observers may include any stakeholders of the project. Any person who has an interest in the test and in the product being tested can participate as observers.
- ***Users/Participants:*** They are a representative sample of people who use the system and are selected to test it. They may be direct users (generally called end-

users) who use the system to complete their tasks, or indirect users who use it for other purposes such as system administrators, installers or demonstrators. They are the most important actors of the user-oriented testing. They perform the tasks handed to them by the evaluator and provide their comments.

- ***Designers and Developers:*** They are the expert in installing, customizing and developing the target product of the test. They usually participate in the test as observers or evaluators.
- ***Stakeholders:*** Those who are affected by the system, or can influence its development, such as the marketing staff and purchasers. Their input is mainly used as additional test results. For example, the marketing staff may like to add or make a specific function always available for any type of users.

Figure 3-5 presents a business case diagram that illustrates the interaction of the different actors involved in the different steps of usability testing process. Usability testing is a high level use case, and ten other use cases are included in it (see the arrows with dashed lines.)

- The Manager/Coordinator is essentially involved in the Plan step. He coordinates the task of the usability team members throughout the process.
- The Monitor is the technical expert and therefore is needed for the Design, Setup, Preview and Conduct steps.
- The Recruiter acquires the Participants.
- The Evaluator attends Preview, Conduct, Compile, Analyze, and Report steps. He is responsible of running the test and reporting the results.
- The Observer (Designer, Developer or stakeholder) helps in the Conduct, Debrief and Report steps.
- The Data Analysts like the Evaluator attend Preview, Conduct, Compile, and Analyze steps.
- The Participants are involved during the Acquire, Conduct and Debrief steps.

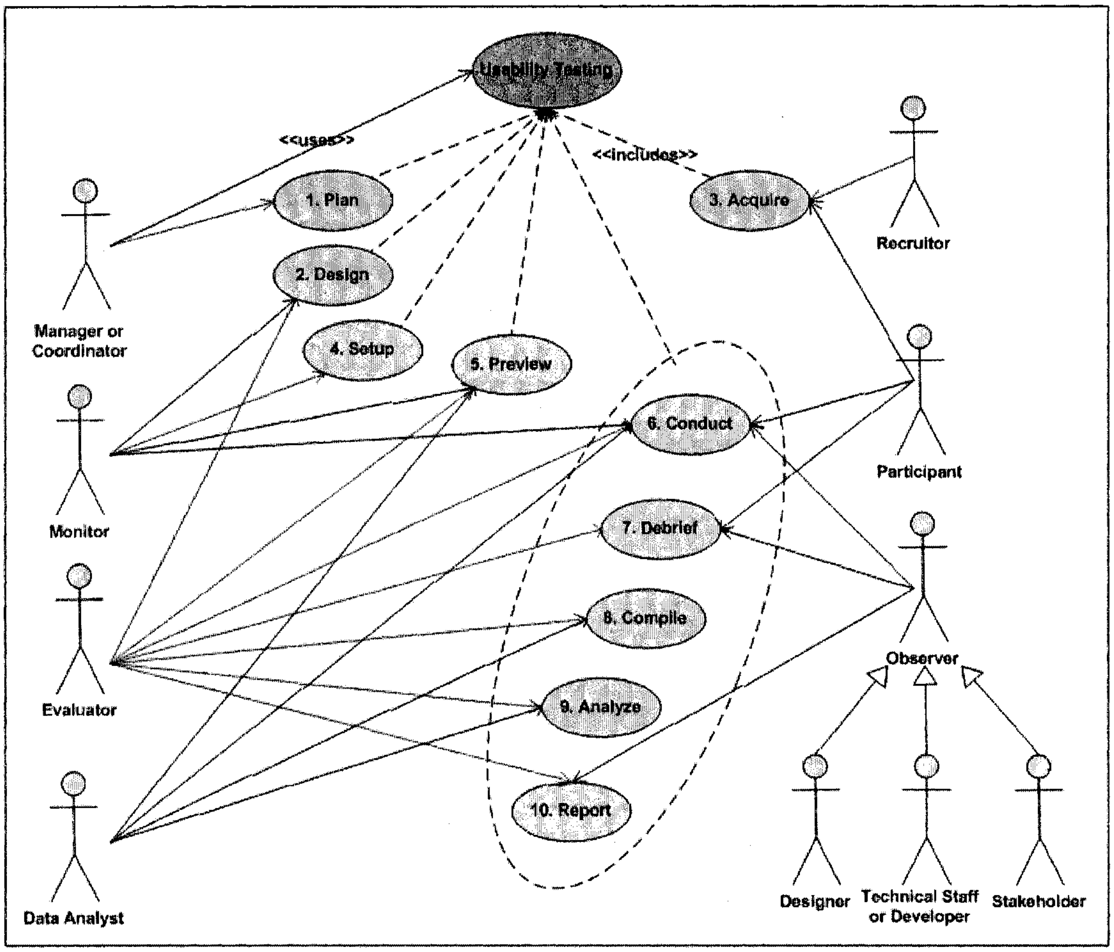


Figure 3-5: User Roles in the Usability Testing Process

Different people's involvement can both contribute to discovering and preventing possible errors and usability issues, and may provide a new approach to understand the product. Most of the time, testers become too familiar with the software, and they may not report a possible error or user problem issue. This kind of situation can be also explained by the fact that developers may be protective of their software, considering their product is bugs or errors free. Introducing user testers and usability professionals into the test team will bring more objective views and alleviate these types of problems.

3.5. Process Workflow

After having defined the steps and activities of proposed usability testing process, it is time to demonstrate how they work together, their logic sequence. Figure 3-6 presents the overall workflow of proposed usability testing process in step level. Most steps have a linear sequence except the *Acquire* and *Setup* steps that can be done in parallel while others are performed sequentially. The decision-making boxes enable to check if the previous steps have been performed correctly, otherwise it is required to go back to the previous steps to fix the issues.After the *Preview* step, it is possible to go back to the *Acquire*, *Setup* and even *Design* steps to fix what have been found missing or incorrect. The *"Conduct-Debrief"* step pair is in some kind of a loop since there is usually more than one participant attending the test.

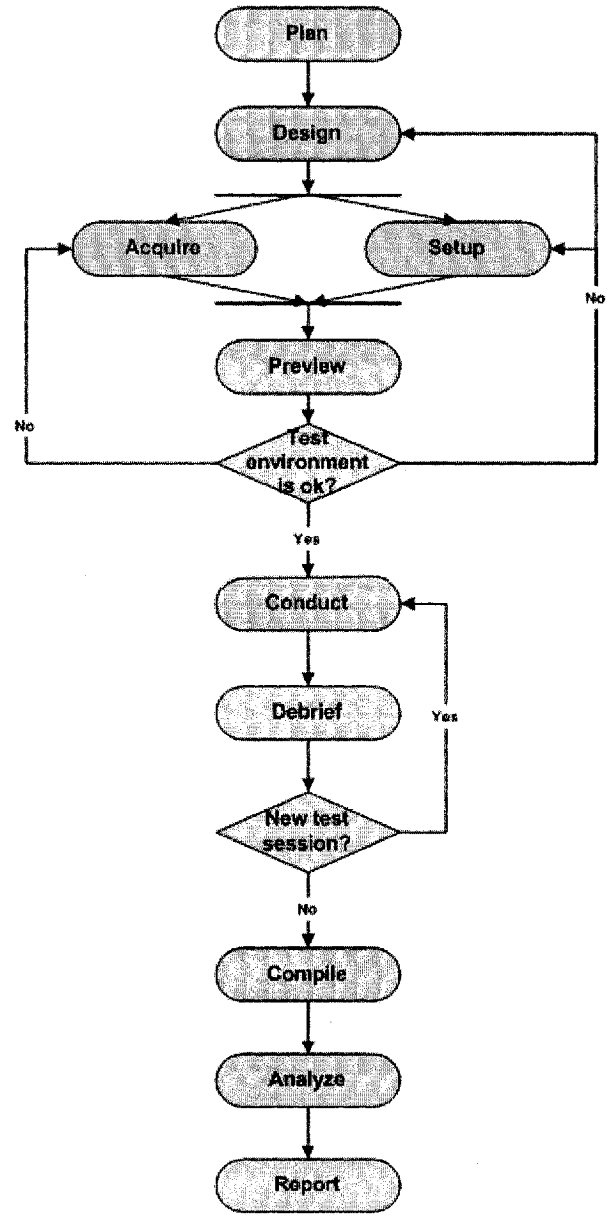


Figure 3-6: Step-Level Workflow

Figure 3-7 is the flowchart in activities level. Each step contains major activities that need to be performed in this step. Within steps, activities are performed in linear sequence. Acquire and setup steps can be done simultaneously after design step is accomplished. Same as in step-level workflow, there are two decision-making boxes. One is between preview and conduct step, and the other is between debrief and compile steps.

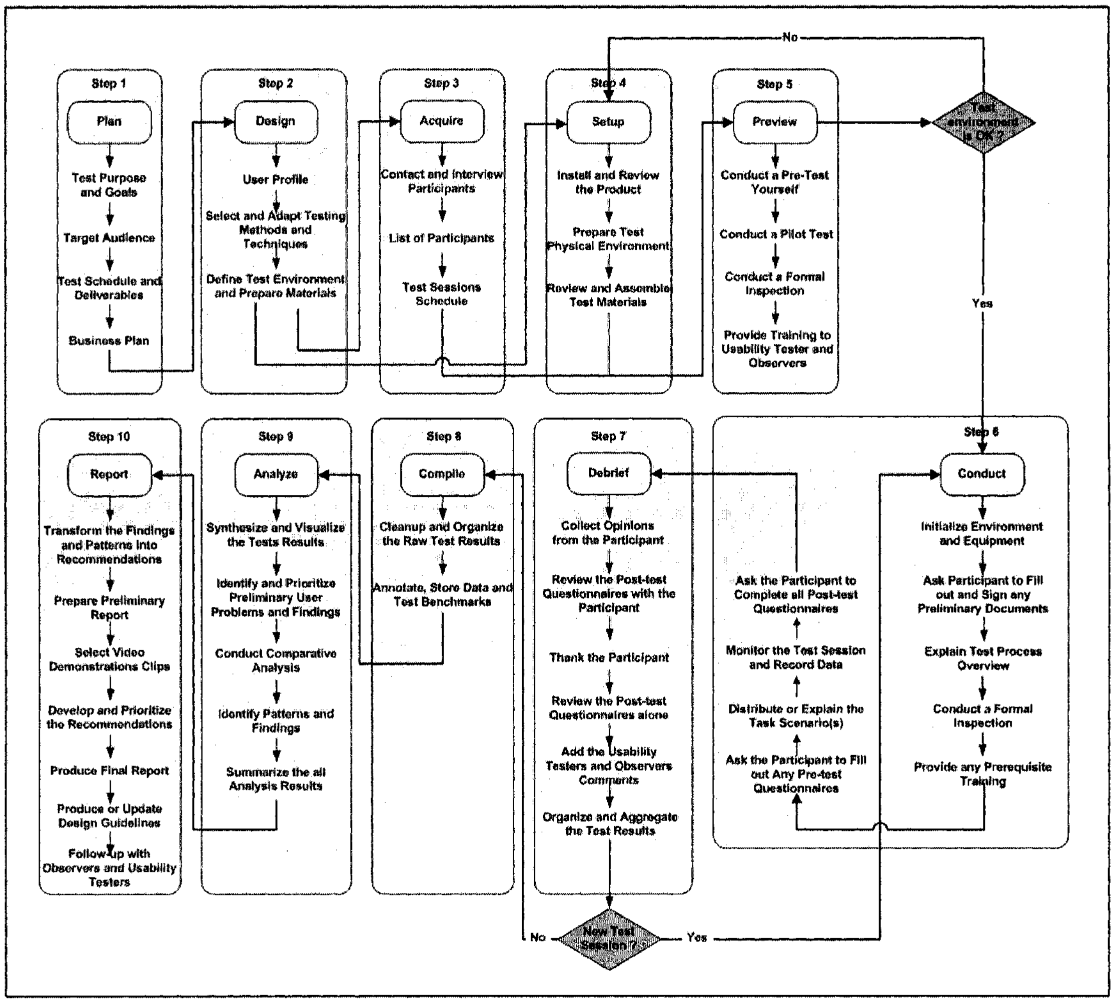


Figure 3-7: Activity-Level Workflow

This workflow answers the *when* part of the question: *who is doing what, when, and how.*

3.6. Summary

In this chapter, a new usability testing process is proposed. As the basis of our process, the common steps, activities as well as templates and methods in the current usability testing processes were kept. Several new milestones that are missing in current processes are added, including hiring participants, conducting pre-tests, and compiling usability data. Moreover, nine usability actors and their roles are defined. These actors are assigned to steps and activities. Furthermore, usability tools are tightly blended with process to bridge the existing gap. To some extent, the proposed process model can be considered as an assembly-line, which gave a clear answer for the key question of process modeling: *who is doing what, when, and how.*

4. WizUse: Design and Implementation

4.1. RANA Architecture

4.1.1. Introduction

RANA (Remote Architecture for Net-Based Analysis) is an ongoing research and development project administered by the Human Centered Software Engineering (HCSE) group of Concordia University. It is an integrative process-sensitive software infrastructure for conducting remote usability tests and user-centered empirical studies.

RANA project aims to provide a platform to access a set of tools via Internet for capturing, visualizing and analyzing the results of empirical studies and usability tests. This infrastructure has the capacity to support some usability tools and offers an architecture adapted for conducting remote usability tests. Ultimately, RANA will provide a web-based and adaptable interface to a toolbox for empirical studies, which will include:

- Video-conferencing and groupware
- Applications for virtual focus groups
- Online participatory design workshops
- Web surveys and interviews
- Remote field observations
- Performance testing

The fundamental objective of RANA is to provide a Computer Aided User Testing Environment (CAUTE). A CAUTE provides the same functionalities to usability and human factors professionals as a CASE tool provides to software engineers. It is intended to offer full assistance throughout the entire user-centered empirical studies process. CAUTE allows repetitive, well defined activities to be automated, thus reducing the cognitive load of the usability and empirical software engineers involved. They are then free to concentrate on the specific aspects of the testing process.

In long term, RANA will have to provide an unified access to a large variety of usability engineering tools including tools developed within Human-Centered Software Engineering lab : MOUDIL (Montreal Online Usability Digital Library), an online library of usability patterns and QUIM (Quality in Use Integrated Measurement), a large database of usability metrics.

In summary, the implementation of the RANA platform is divided into 3 phases:

- Phase one – Design a generic usability testing process.
- Phase two – Design an architecture to support remote usability testing; design and implement a web-based wizard to assist usability professionals managing and conducting usability tests.
- Phase three – Integration of usability tools such as QUIM Editor, MOUDIL, etc. Detailed descriptions of these tools are provided in appendix A.

WizUse is one of the component tools within RANA platform to assist usability professionals conducting usability tests. WizUse covers the second phase of RANA implementation and I am going to present later in this chapter.

4.1.2. Platform for Remote Usability Testing

In the first stage of RANA project, one of the major tasks is to define the architecture supporting remote usability tests, which combines two types of labs: mobile and fixed together. However, the tests are conducted remotely via Internet. The data analysis power of fixed labs and the flexibility of mobile labs are the advantages we have taken and intend to contribute to this platform. The architecture that I defined is mainly made up of four components (Figure 4-1): (1) user's local environment, (2) observers' remote environment, (3) mobile usability lab, and (4) fixed usability lab.

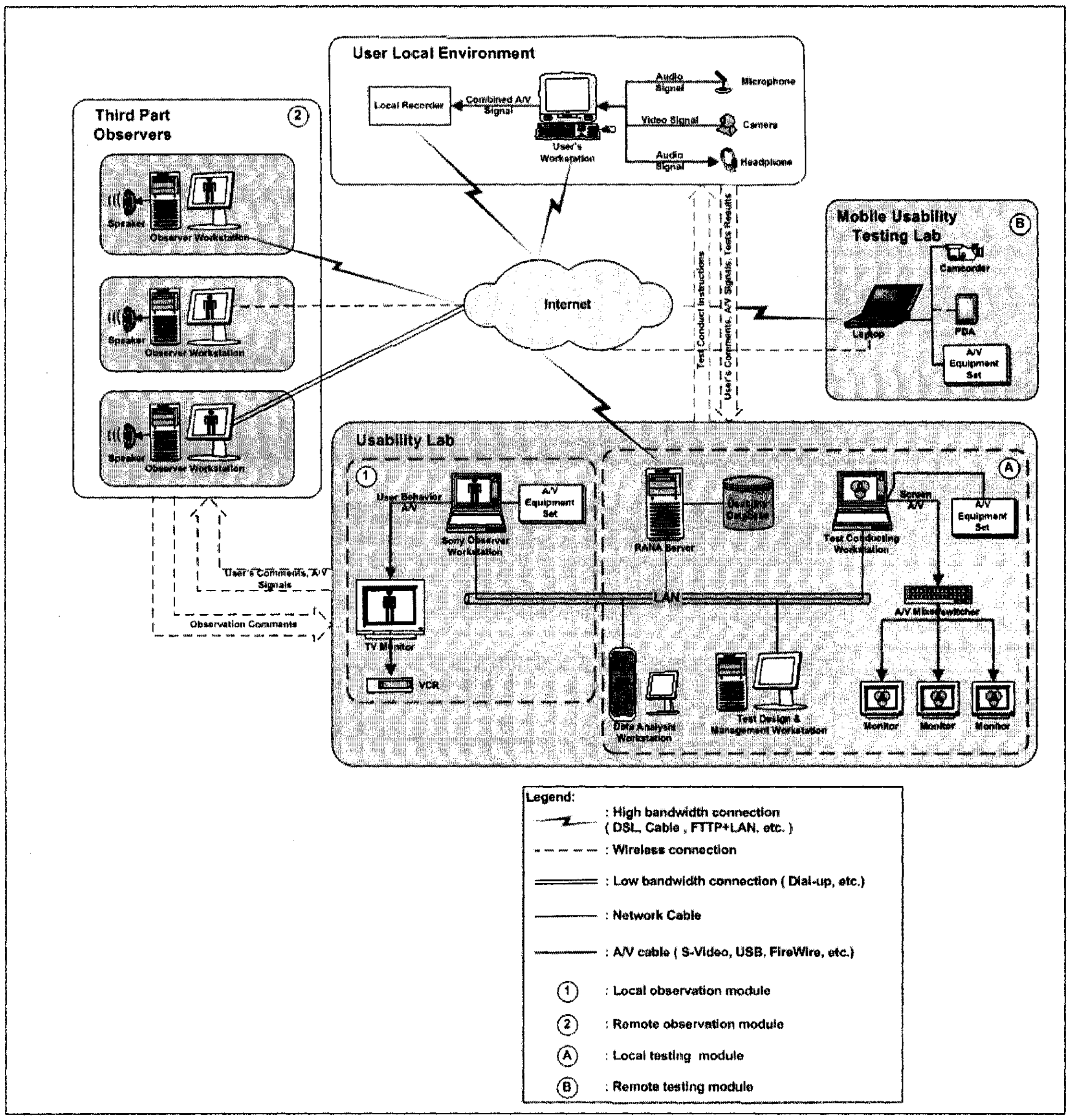


Figure 4-1: Architecture for Remote Usability Tests

The user's local environment includes user's workstation and minimal audio-video equipment, a web cam, a microphone, headphones and/or loudspeakers. This audio-video equipment is necessary for collecting the multimedia data, including user's behaviors and interactions with the computer. It also ensures an interactive communication between the people involved in the tests: the testers, the observers and the participants. It is better that the participant has a higher band-width Internet connection in order to facilitate the real time A/V data exchanging and the data transferring.

The observers can be located in mobile or fixed labs or dispersed geographically. In addition, no constraint is made concerning their connection speed, meaning that they can have a low or high band-width connection.

The mobile usability lab is made up of a laptop that has a wireless Internet connection. Besides the laptop, a PDA, a camcorder and an audio-video equipment set, including a web cam, a microphone, headphones and/or loudspeakers, are required.

The fixed lab is composed of two modules: the observation module and the test module. The main function of the observation module is to enable observers to record and visualize participants' behavior during the test through a big monitoring screen. The test module is equipped with a workstation and three monitors. A switcher enables to combine the different sources of audio-video signals, and to project them on the three monitors. One of the monitors is used to display the participant's screen, and two others are used to display the participant's behavior and face. The main function of this test module is to provide an adapted infrastructure for the test supervision and the interaction with the participant.

In the center of this architecture, the RANA server plays a double role:

- As a web server: manages usability tests, facilitates the communication between different actors involved in usability tests via Internet.
- As a data server: stores both audio-video and textual generated data by the usability tests. Testers can not only find raw data here, but also analyze them.

This server is beyond the traditional web and data storage server by providing multimedia functionalities. It exploits a representation of all observation data based on MPEG 4, 7 and 21 norms [Bormans et al. 2003;Nack and Lindsay 1999] in order to avoid multitude data conversion.

The proposed environment enables to perform synchronous and asynchronous tests. During synchronous tests, the testers, the participants as well as the observers are connected to each other at the same time, and the test visuals are shown in real time to all people involved in the tests. In that case, the participant and the testers interact via Internet by video conferencing and remote control tools, such as Microsoft NetMeeting, and Timbuktu. The testers use remote control, whiteboard and file sharing functionalities for distributing the scenarios and observing the participant's screen. The observers cannot interact with participants, but observe them in real time.

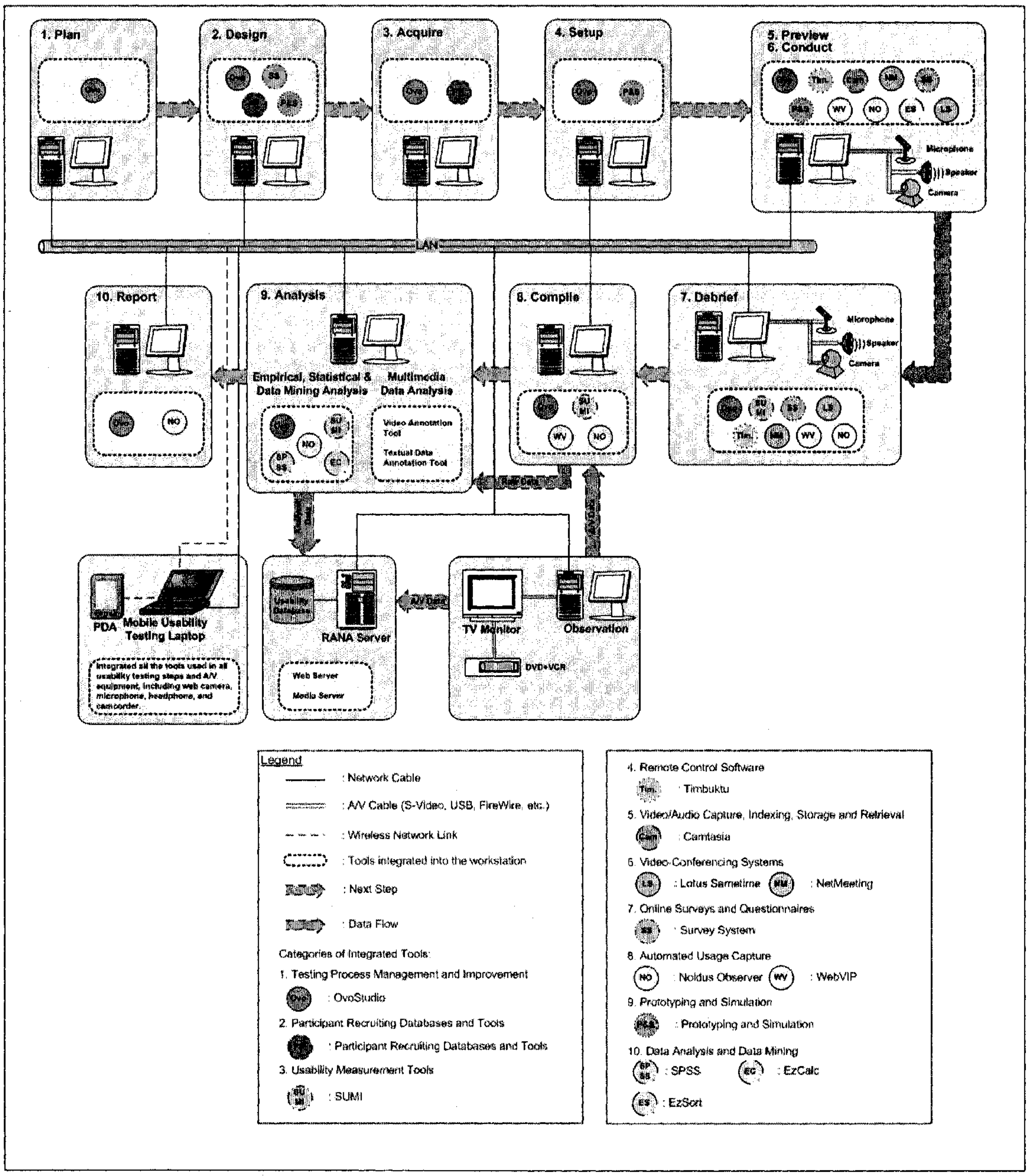


Figure 4-2: Magnified Structure of Local Usability Lab

During asynchronous tests, participants perform the tests whenever they want. The testers and the observers can get back collected data and the results afterwards. Data collecting task can be done automatically by an agent on the server side or on the user's computer. [Hilbert and Redmiles 1998]. The observers provide their comments and feedbacks by email or via RANA website, in real or postponed time. Collected data can be accessed via the RANA server by all usability team members involved in the test project. In usability lab, the testers will analyze video and textual data and annotate them by using a video annotation tool. Observation patterns can be discovered and stored into the usability database.

In figure 4-2, I have magnified the structure of remote usability testing platform in local usability lab, by integrating our proposed usability testing process and the tools selected from our tools survey. These tools are classified into ten categories that have presented in usability tools section of chapter 3. These representative usability tools are distributed into testing steps where they can be deployed. This figure also shows the logical process sequence and data flows within the process. In order to demonstrate the structure more clearly, I put one desktop in each test step, however, in reality we do not need to provide desktops for each step since hardware equipment can be reused.

4.2. WizUse: Introduction

WizUse is a web-based usability testing wizard for supporting usability tests, which provides a platform to visualize usability-testing process and manage a usability knowledge base that can assist usability testers to accomplish their tasks easily. WizUse can be integrated into a CAUTE as a fundamental tool.

Again, a CAUTE provides the same functionality to usability and human factors professionals as a CASE tool provides to software engineers. It is intended to assist the entire user-centered empirical studies process. CAUTE allows repetitive, well defined activities to be automated, thus reducing the cognitive load on the usability and empirical software engineers involved. They are then free to concentrate on the creative aspects of the testing process.

Some tools are integrated to support particular methods, reducing any administrative load associated with manually applying those methods. Like those tools, CAUTEs are intended to make usability testing and empirical studies more systematic, and they vary in scope from supporting a specific step in the testing process to encompassing the complete life cycle.

There are four main functionalities that WizUse can offer:

1. ***Usability testing process visualization***:

   WizUse visualizes usability testing processes and presents the processes as a four-level structure:

   - ***The steps:*** that were defined previously: plan, design, acquire setup, preview, conduct, debrief, compile, analyze, and report. This well structured and ordered process in steps enables usability specialists to organize and follow up correctly the whole life cycle of the tests. It also gives the outputs and the artifacts used for each step. Furthermore, testers can always be aware that in which step of process they are, and the status of the process. For example, which steps are accomplished and which ones are needed to be done.

   - ***The actors:*** involved in each step, their roles and responsibilities. The different actors are managers/coordinators, monitors, recruiters, evaluators, data analysts, observers, users/participants, designer/developers, and stakeholders. The classification in roles helps to identify the skills required for each type of actor, and then, a set of tasks is assigned to each actor. The wizard highlights the actors involved in each step and also provides detailed description of their roles and responsibilities in the current step.

   - ***The activities:*** describe the tasks required for each step. For example, in the plan step, the activities defined are the following: (1) define the test purpose and goals, (2) define the target audience, (3) define the target product, (4) define the test schedule, resources and outcomes. Thus, the usability tester knows exactly what has to be performed for each step and in which order.

   - ***The methods and tools:*** WizUse manages a knowledge base of relevant methods and tools used by the usability professionals. For each step, the wizard suggests the most suitable tools and methods for well conducting the tests. In the knowledge base, useful templates are also provided. WizUse also suggests the deliverables that are needed to prepared, which give usability testers a clear idea what the output they are going to achieve.

Thus, WizUse makes the whole usability testing process more formal, automatic and visible.

2. ***Usability testing process management***:
   WizUse creates a knowledge base that stores all usability testing process information that is stored in the four-level structure addressed above. Users can add their own processes, actors, activities, methods and tools in the knowledge base to enlarge it. By doing so, users experience, research and practice can be accumulated.

3. ***User privilege management***:
   In WizUse, users are classified into different groups, and assigned to predefined roles, and given different levels of privileges. These privilege levels determine the user's access capacity to the knowledge base.

4. ***Usability testing project management***:
   In this function level, WizUse is something like an Information Management System (MIS), which allows usability professionals to plan and design their usability tests by filling online pre-designed forms, automatically generate testing plan and design documents. Eventually, WizUse will create a final CIF style usability testing report according to the information provided by the test evaluator. Meanwhile, the wizard offers a unified interface to let the user access usability tools that have integrated into the RANA server.

In the following sections, I am going to present WizUse in detail from two aspects: design and implementation.

4.3. Why a Wizard?

4.3.1. What is a wizard?

A wizard is a structured series of dialogs that lead a user through a task one step at a time, asking questions and use these answers or choices to achieve a goal or accomplish a task

[Scanlon 1997;Sun-Microsystem 2001]. Wizards provide all required information needed to perform the task, and await users' responses, including keyboard entering and the choices that the user made. Typically, wizards are intended to simplify a task so that inexperienced users can perform it easily, or to expedite a complex task by grouping its steps in a single place, or to do both. [Sun-Microsystem 2001]

When a task is too complex to fit into a single wizard, a launchpad is needed. A launchpad can be defined as a graphical user interface that is used for too complicated tasks [Wickham 2001]. The launchpad acts as a central access point for launching a series of related wizards, dialogs, or combination of both. We have applied this concept to design the fourth functionality of WizUse. Figure 4-3 gives an example how a launchpad looks like. [Wickham 2001]

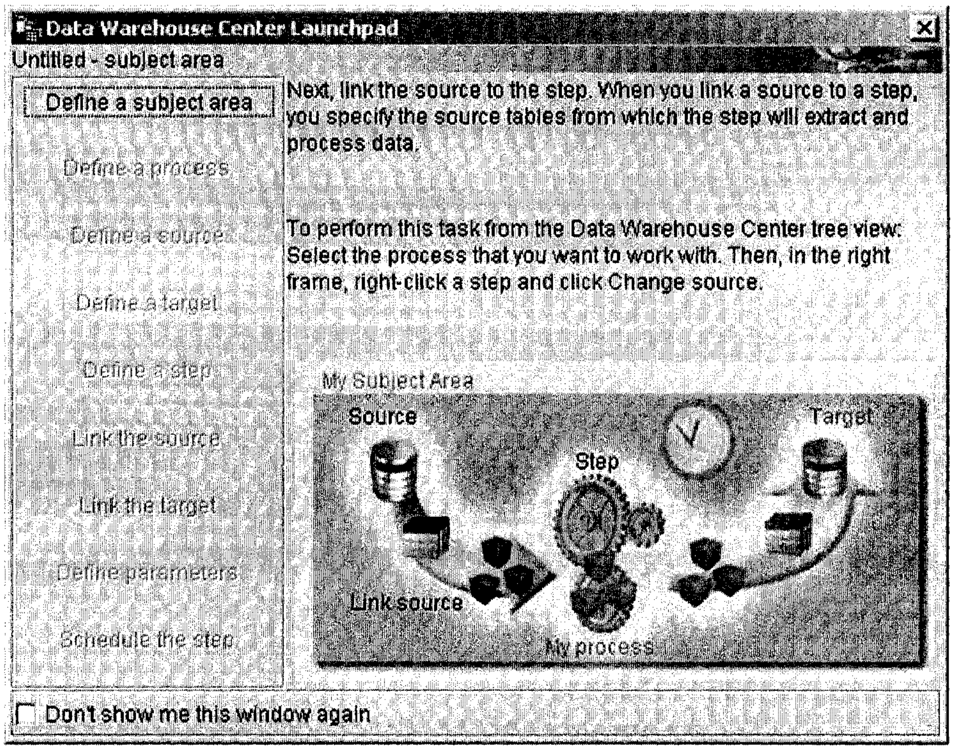


Figure 4-3: An Example of Launchpad

4.3.2. When to consider a wizard?

Since creating a wizard is not always the best solution for some tasks, it is better for us to consider when wizards can help. In the following three situations, we need to create a wizard.

1. The goal that users want to achieve has many steps. Simple tasks do not require wizards. When a task is very complex and needs many steps to accomplish it, under such a situation, in most of the cases a wizard is needed. A wizard decomposes the complex task into steps, assists users to perform one step at a time, and guides them to go through the whole process to get the task done.
2. Users lack the necessary domain knowledge. In some cases, when a novice wants to perform some tasks in the domain they are not familiar with, it is better to provide them some expertise from experts in this domain. Wizards can work very well in this circumstance. Wizards supply an expert's perspective in an area that is beyond users' capacities.
3. Users must complete a task in a particular sequence. Some tasks always have a specific process to accomplish. Wizards are often useful in this type of task to make sure all steps are performed in the proper order.

In our case, the task of conducting a usability test is qualified for all situations presented above, that is test conducting is a complex task that needs a set of steps to perform in a certain sequence. Some usability testers are novices and lack required knowledge in usability domain. Designing a wizard to give some assistance is the best solution.

4.3.3. How to design a wizard?

According to Scanlon [1997] and Bollaert [2001;2002], we need to keep in mind the following issues:

- Roadmaps: successful wizards show an overview of the functions they contain and where they appear. A roadmap is needed to clearly mark the boundaries of the wizard. Users should be always informed where they are and what they are doing.
- Clarity of Inputs: at every step, a wizard should provide enough information for users to make decisions and interact with them. Clear and complete descriptions will reduce the chance that users go to help section.
- Predictability of outputs: users need to understand the implications of each choice they make in the wizard, and need to know what the results that wizard is going to generate.

- Avoid long screen that require scrolling: wizards are meant to be fast and easy. For this reason, when we design interfaces of a wizard, we should always keep in mind to put the most important contents in one screen without scrolling down.
- Design flexible navigation: always give users a safe exit when they do not want to complete the whole process of the wizard. Give users facilities, such as next and previous buttons, to access next and previous steps. It is essential that we need to maintain the data users have already entered when they go back to the previous steps. Users will be very frustrated if previous entered data disappears.
- Provide shortcuts or accelerators: since wizards are not always the best user interface for expert users, it is reasonable to provide some shortcuts or accelerators for experienced users, letting them customize the process of tasks. Another reason to offer shortcuts is that novices will be familiar with the wizard interface and become experienced users.
- Consider security: in some cases, information is accessible by authorized users. That means we should provide a secure login/logout procedure.

Points above are the guidelines, according to which we design WizUse interfaces.

4.4. Conceptual Design

Fundamentally, WizUse is an integration tool, which integrates usability processes, tools, methods and templates while helping to design user-customized usability testing processes. Figure 4-4 shows the concept architecture of WizUse.

Firstly, usability professionals select the testing process, methods, tools and template they prefer, and then combine all these components together to create their own customized usability testing process, which will be stored into the usability knowledge base. They can modify the process at any stage of their testing process.

Secondly, after customized process is generated, testers have two options. One is to use process visualization component of WizUse to visualize process and use it as a guideline to assist their testing. All actors, activities, usability tools, methods and templates are distributed into step level to let testers know who is doing what, when, and how. The other option is that testers can use project management functionalities to manage their testing projects and store the related data into usability database. The database provides a controlled access system by predefined privilege settings.

Finally, testers can generate the standard formatted usability testing report by retrieving data from usability database.

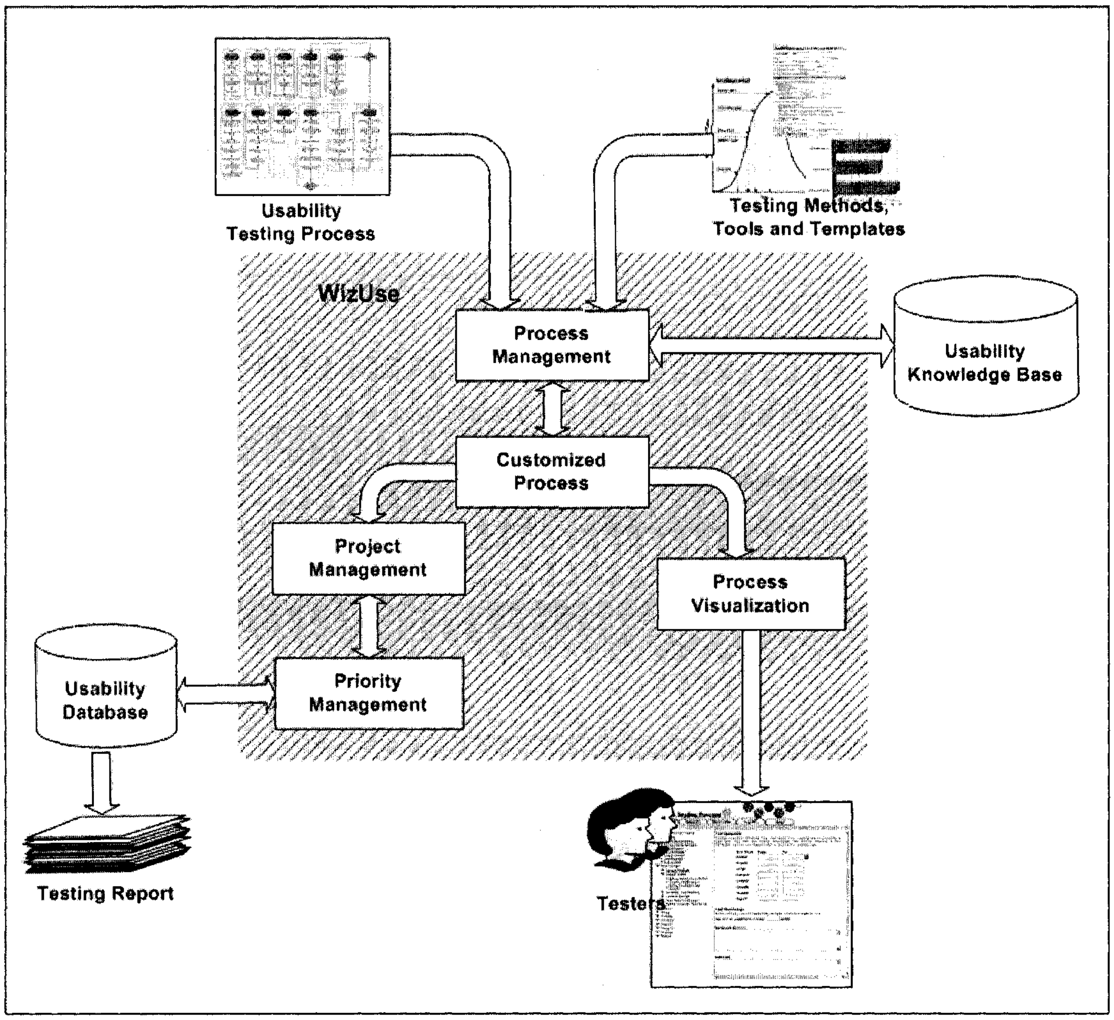


Figure 4-4: Architecture of WizUse

4.5. WizUse Implementation Stages

According to Amsden [2001] and Sharma et al. [2002], there are four levels of tool integration:

- Process level integration: Invoke or launch a tool on resources of a specific type in a separate window.
- Interface level integration: UI integration allows a tool to participate with other tools as if they were designed as a single application. (Same look and feel)
- Data level integration: Data integration allows data manipulated by the tool to be available to other tools.
- API level integration: API integration allows client applications to access data through tool-specific client APIs.

Based on tools integration levels, we divided WizUse implementation process into two stages:

- The first stage: Process level integration. In this stage, there are two phases. The first one is that we provide the same style of interfaces to visualize the processes, methods and tools. For every component, we provide the link to either invoke or download. Users can also add in their own processes, methods or tools. The second phase aims at automating usability testing process by adding some automation functionalities, such as test projects management, report generation etc. Testers can follow our predefined testing process and manage their test related information by filling web-based forms. Moreover, we combine selected usability tools into the WizUse. Thanks to these tools, they are making some steps fully automatic in the process. For example, Morae makes the conduct step totally automatic.
- The second stage: Data level integration. This stage intends to offer an inter-medium data structure to allow usability tools to share their data.

In this thesis, we only cover the first stage that we have finished implementing.

4.6. Architecture Description

There are several available options for languages, tools and database server when implementing web applications. The choices made for WizUse mostly base on the development team skills and expertise, the available material at the Concordia Usability and Empirical Studies Lab, the current practices in the industry, the portability and maintainability of the application, the timeframe, and finally the costs. Table 5-1 describes the development and application running environment.

Development Environment	**Description**	**License**
Development Platform	Windows XP	Not Free
Development Tool (HTML)	Macromedia Dreamweaver MX 2004	Not Free
Database Design Tool	DBDesigner4	Free
Languages	PHP 4.3.8, HTML, XML, JavaScript	GPL
Report Generator	PDF Writer (Embedded with PHP)	Free
Control Version Tool	Share360 Groupware v.2	Not Free
Production Environment	**Description**	**License**
Production Platform	Red hat Linux 8.0 3.2-7 Kernel version 2.4.18-14smp	Free
Web Server	Apache 2.0.50 with PHP handler	Free
Database Server	MySQL 4.0.20	Free
Administration Tools	PHP MyAdmin 2.5.7	Free
Hardware	Two 1266 MHz Pentium III processors, 1.2 GB RAM, 16.6 GB hard drive	--

Table 4-1: WizUse Development and Running Environment

The development process will be a standard 3-tier process following MVC (Model View Controller) design pattern [Leff and Rayfield 2001] to ensure robustness, good maintainability and enhanced portability. In a 3-tier application, the Data, Business and Presentation components are all separated and can even be on different machines, thus offering maximum flexibility.

The system architecture is pretty much alike any existing web application. At one end, a MySQL Database server handles requests from the Apache Web server through the PHP functions calls, and at the other end, users interact with dynamic web pages following

HTTP requests to the web server. The MySQL database server and the Apache web server can run on the same or different server. In our case, we bind them together in RANA server. Figure 4-5 depicts the interactions between the user, database and web server. In the figure, in order to demonstrate clearly, we separate web server and database server apart.

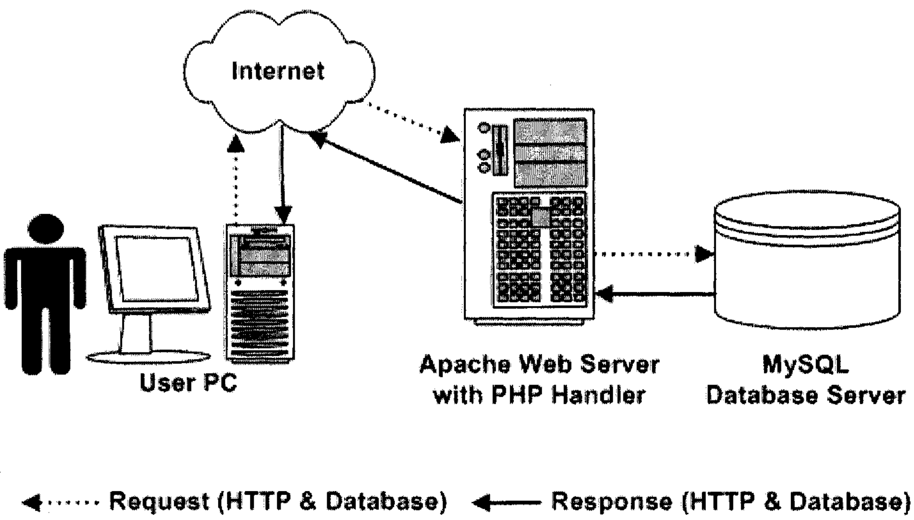


Figure 4-5: Application Architecture Diagram

Logically, the application structure could be viewed as three layers, from top to bottom, presentation layer, business layer and data layer (Figure 4-6). End-users only interact with the presentation layer as business and data layers are hidden from them. The business layer works as middle layer between presentation and data layer. The business layer passes the requests from the presentation layer to the data layer, and in the meantime it retrieves data from the data layer and transfers the data to the presentation layer.

In the database layer, we have defined thirty tables. Figure 4-7 shows a portion of WizUse entity relationship diagram (ER diagram). This diagram focuses on the core entity, tests, and other entities that have relationship with it. In WizUse, usability tests are the centre of the whole data structure, they have the relationship with most of other entities, such as participants and tasks etc.

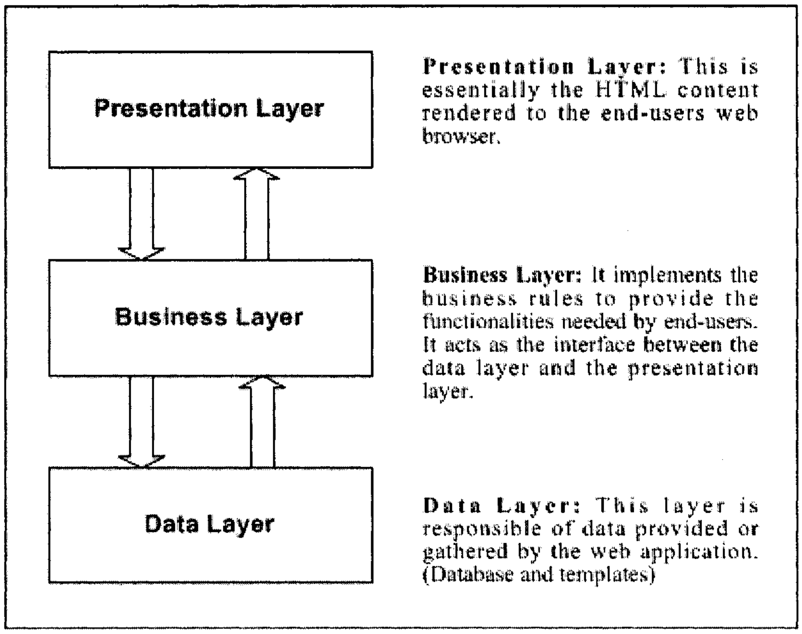


Figure 4-6: Application Logical View

Database structure

It is true that some of entities can be integrated into test entity, but we choose to separate them apart for the efficiency and flexibility. We create usability processes, methods, tools and templates as independent entities. They are the components that can be used for any usability test. When users of WizUse create a new test project, they can either create new components or reuse existing components by adding relationship. When they want to modify their project, the only thing they need to do is to change the relationship between test entity and component entities. They do not bother to be in charge of maintaining component tables. Thus, the efficiency and flexibility of the application are increased. The whole view of ER diagram can be seen in Appendix D.

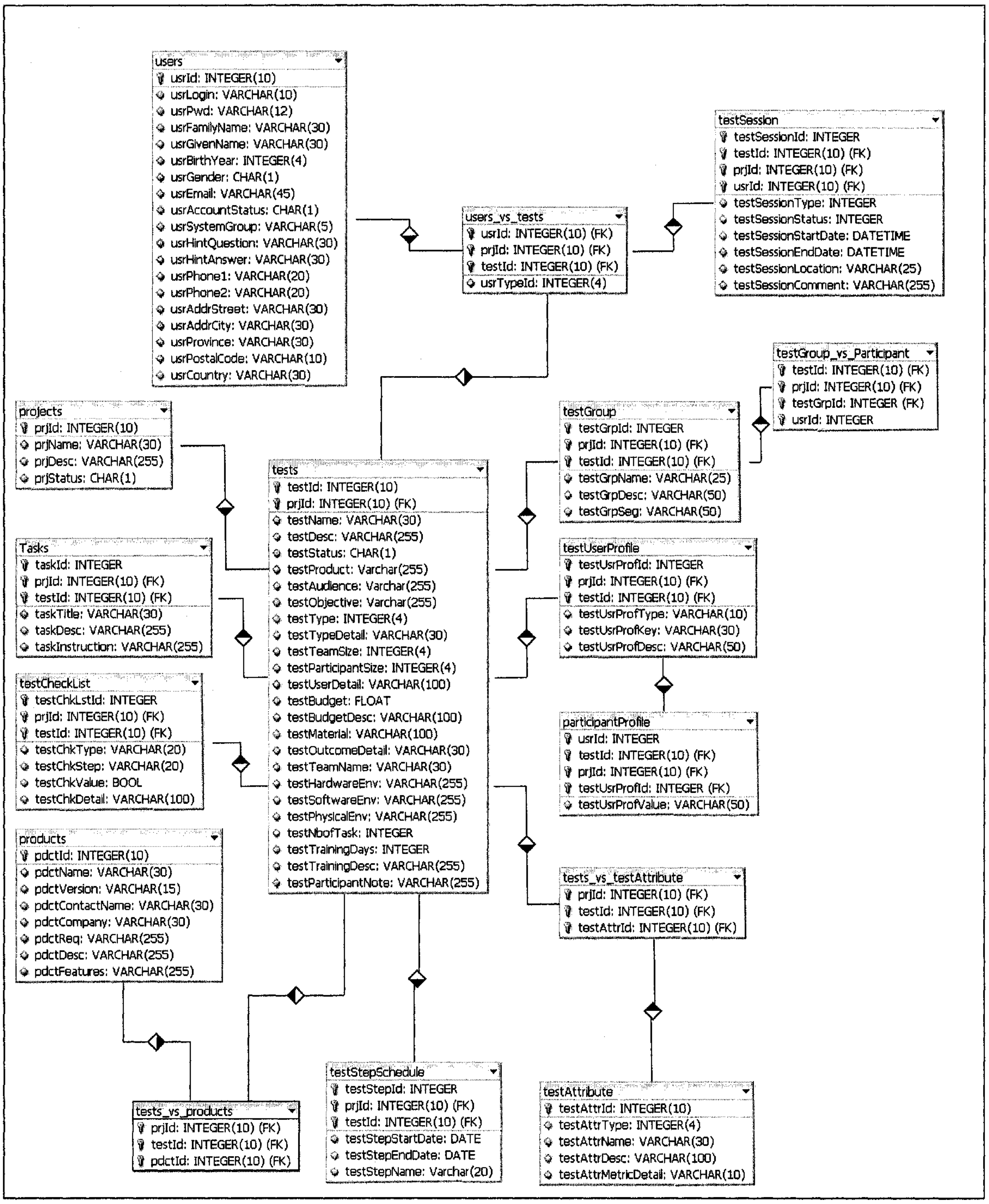


Figure 4-7: A Portion of ER Diagram

4.7. User Interface Design

The user interfaces of WizUse have been designed based on clearly stated guidelines. The screen (Figure 4-8) is divided into 2 major parts:

- A top section where the banner and the Function Panel are located
  - The Function Panel gives access to the main functionalities offered by the application
- A bottom section is also divided into 2 areas, the left panel and the right panel
  - The left panel is the usability testing process. It is designed as tree structure and users can switch between different steps by clicking the step title.
  - The right panel is the place where users enter the data. It begins with a title that identifies the usability testing process and which matches the step name on the left panel. Below the title, there are the forms used to collect the data followed by the commands buttons. Finally, a navigation bar ends the right panel and is used to switch between collections of records.

Other considerations about HTML Design

- Define a global style sheet (.css) file
- Build a library of JavaScript functions for form's input validation
- Avoid using client-side JavaScript, use server-side JavaScript instead

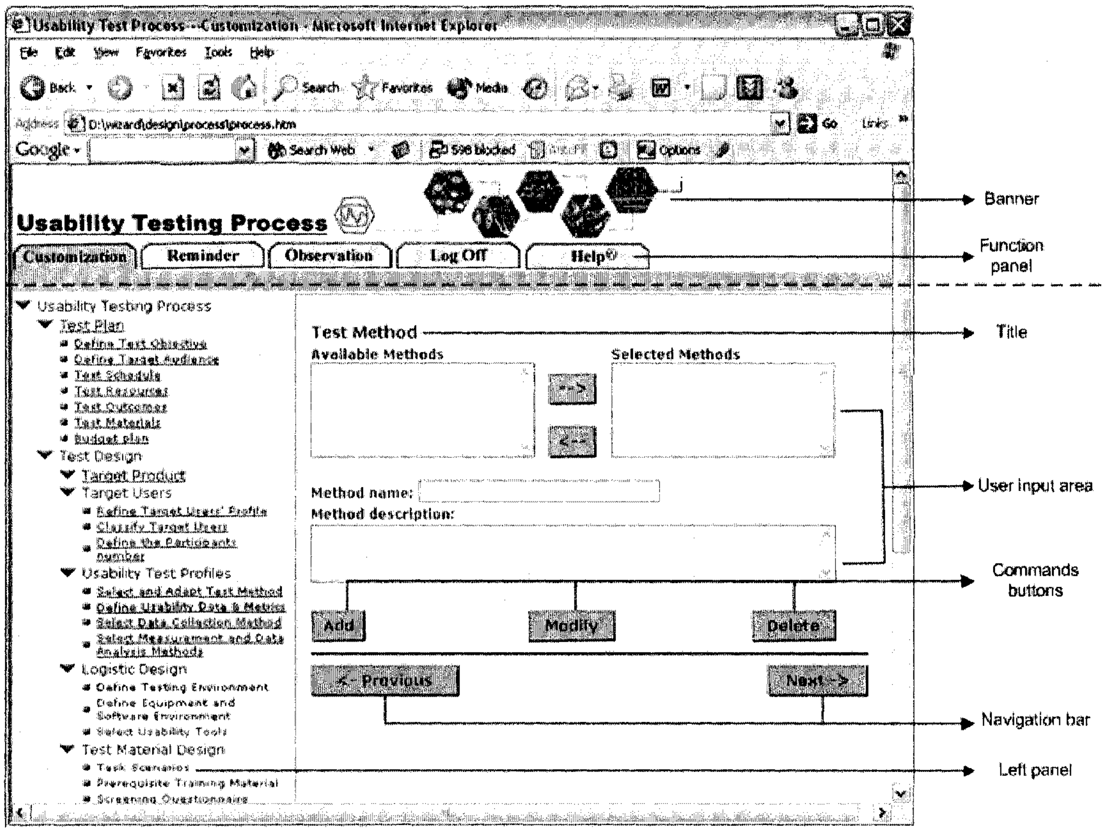


Figure 4-8: User Interface of WizUse

4.8. Functionalities Description

Since there are two phases in the first implementation stage of WizUse, I am going to present application functionalities according to these phases. In the following three sections, I will describe functionalities in the first phase. After that, I will present the functionalities that provide in the second implementation phase.

4.8.1. User Registration and Management

User Registration and Account Management functionalities are most fundamental ones within WizUse. They are mostly needed for the second implementation phases where the platform will customize the interfaces and the process according to the user profile. Furthermore, users' profiles determine their privileges for accessibility of testing processes. Table 4-2 shows different kinds of WizUse users (actors) and their privileges. FA means full access, by which users can read, write and modify rights to the usability database. LA refers to limited access that lets users have only read right to access the information related to a specific step of a test project. NA refers to no access, which means the user has no right to access any data. Managers and WizUse administrators are the advanced users who can perform all functionalities, while others are ordinary users who can only perform a certain set of functionalities according to their privileges.

Step / User	Plan	Design	Acquire	Setup	Preview	Conduct	Debrief	Compile	Analyze	Report
Manager	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
Monitor	LA	LA	LA	FA	FA	FA	LA	LA	LA	LA
Recruiter	LA	LA	FA	NA	NA	NA	NA	NA	NA	NA
Evaluator	LA	LA	LA	LA	FA	FA	FA	FA	FA	FA
Data Analyst	LA	LA	LA	LA	FA	FA	FA	FA	LA	LA
Observer	LA	LA	LA	LA	LA	LA	LA	LA	LA	LA
Participant	NA	NA	NA	NA	LA	LA	LA	NA	NA	LA
Developer	LA	LA	LA	LA	LA	LA	LA	LA	LA	LA

Table 4-2: User's Privileges

By default, we set designers and developers limited access privilege to all usability testing process as observers (please see the actors definitions in chapter 3). According to the definition of designers and developers, they can also act as evaluators. That means they have the same privilege as evaluators.

We are using these functionalities in the first phase for authentication purpose and session management.

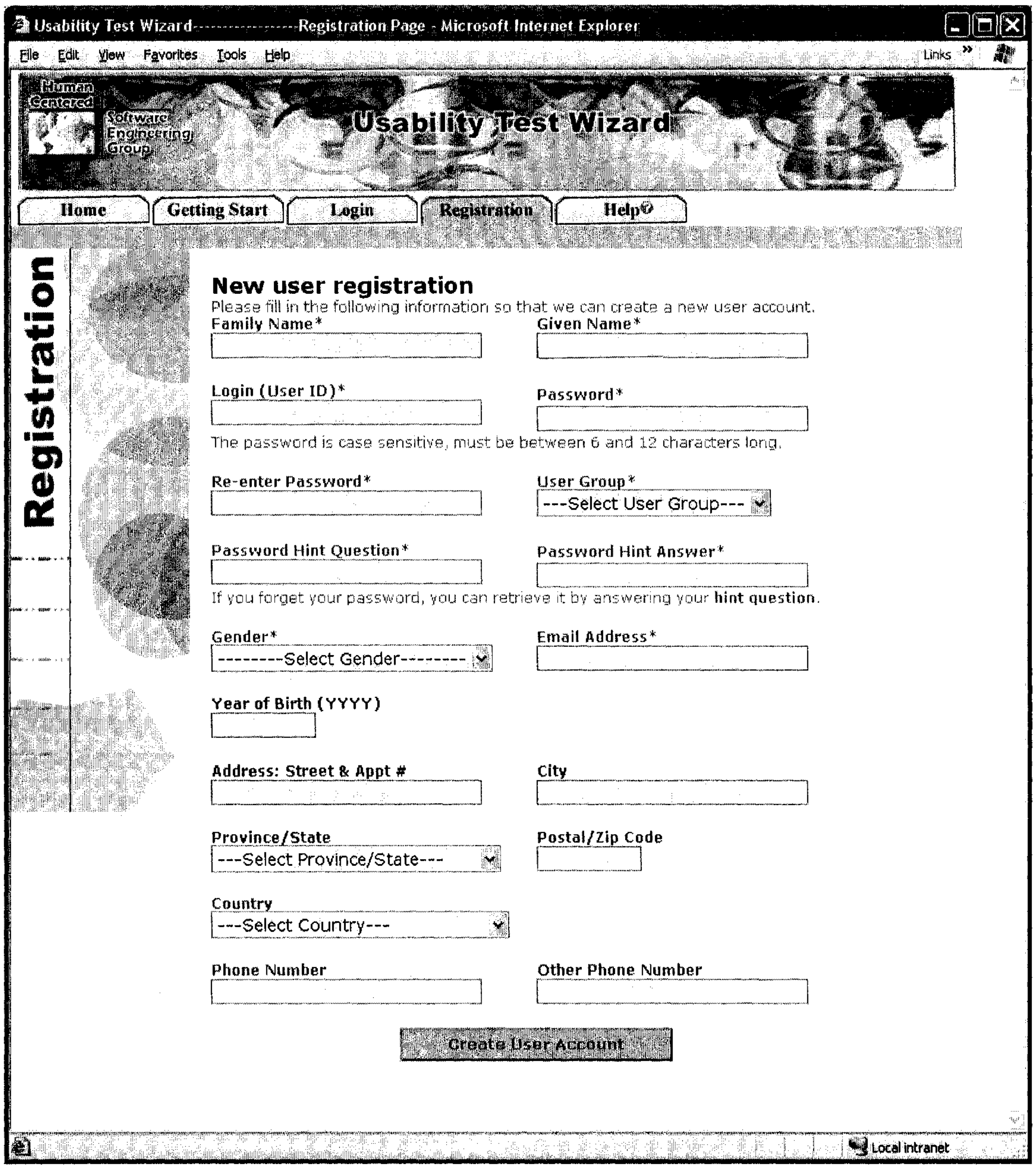


Figure 4-9: Create Account User Interface

For ordinary users, registration and management functionalities include:

- Create User Account. Figure 4-9 is the screen shot for new user registration interface. For the data entries, which are in red color and marked with the star symbol, are required information. User accounts cannot be created without that information.

- Update User Profile. Users can create and modify their profiles, such as address and contact information etc., by performing this function.
- Login / Logout. When users want to use WizUse, firstly they must login the system by providing valid User ID (Login Name) and Password. At the same time their sessions start, and sessions will expire once users logout or after a long period of inactivity.

For advance users, we provide some advance account management functions as follows.

- Manage User's Privilege. Advance users can set or modify ordinary users' privileges by assigning them to different actors. Actors' privileges are defined in table 4-2.
- Lock/Unlock User Account. Advanced users can lock a user when they want to temporarily block him/her from accessing the wizard.
- Delete User Account.

4.8.2. Usability Testing Process Management

WizUse has been developed to fully support the proposed usability process presented in chapter 3 and any other usability process that can be modelled as shown in Figure 4-10 below [Nganou 2005]. That model presents the usability process as a sequence of ordered steps that produce deliverables and are composed of activities executed by actors who use particular methods and tools.

As I have mentioned before, all components related to test process are stored in the usability knowledge base, which is embedded into MySQL database server running in RANA server. Initially, we put our proposed usability testing process as well as the methods and tools we selected from our research into the knowledge base. Users can add their own processes, methods and tools or modify existing components by performing the functionalities addressed in this section. After doing so, a user customized usability testing process comes into being, and then the user can use process visualization functions to visualize the process.

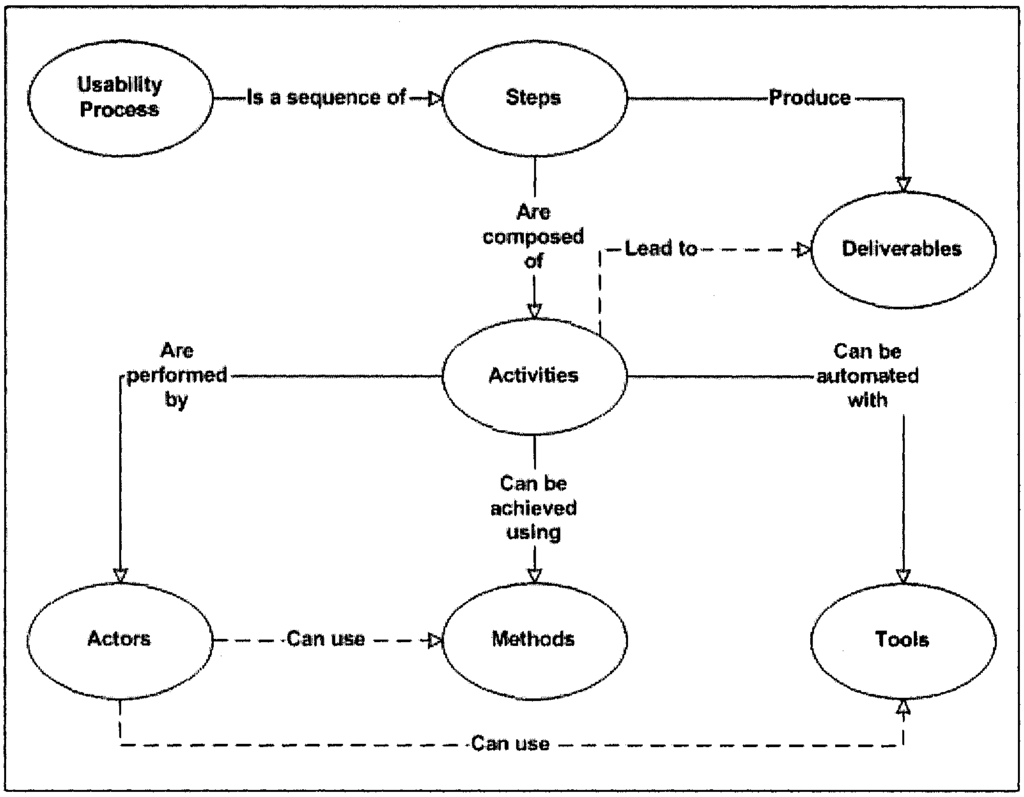


Figure 4-10: Usability Process Meta-Model

Functionalities in this section are used to define, customize and maintain usability testing processes and components that are stored in the knowledge base. They are designed to support the evolution of the process and be compatible with the RANA platform. The functionalities related to process management are:

- Manage Usability Process. It provides the interfaces to initialize a new usability testing process (Figure 4-11), update process information and delete process. A process is defined by its name, version, description and status.
- Manage Usability Process Steps. The user can define, update or delete a process step through these interfaces.
- Manage Usability Process Step Activities. Allow user to create new activities for a process step, update and delete existing ones.
- Manage Usability Actors. It gives the ability to define new actors that can later be assigned to a process, update and delete current ones.

- Manage Usability Artifacts and Utilities. Through these interfaces, the user can define, update and delete usability process components, which include artifacts, deliverables, methods, tools and templates.
- Assign/Remove Actors to/from Usability Process. This functionality is used to define the role of an actor in the steps of the process.
- Assign/Remove Usability Artifacts and Utilities. It helps to define or remove the association between deliverables, methods, tools and templates on one side and process step and activities on the other side.

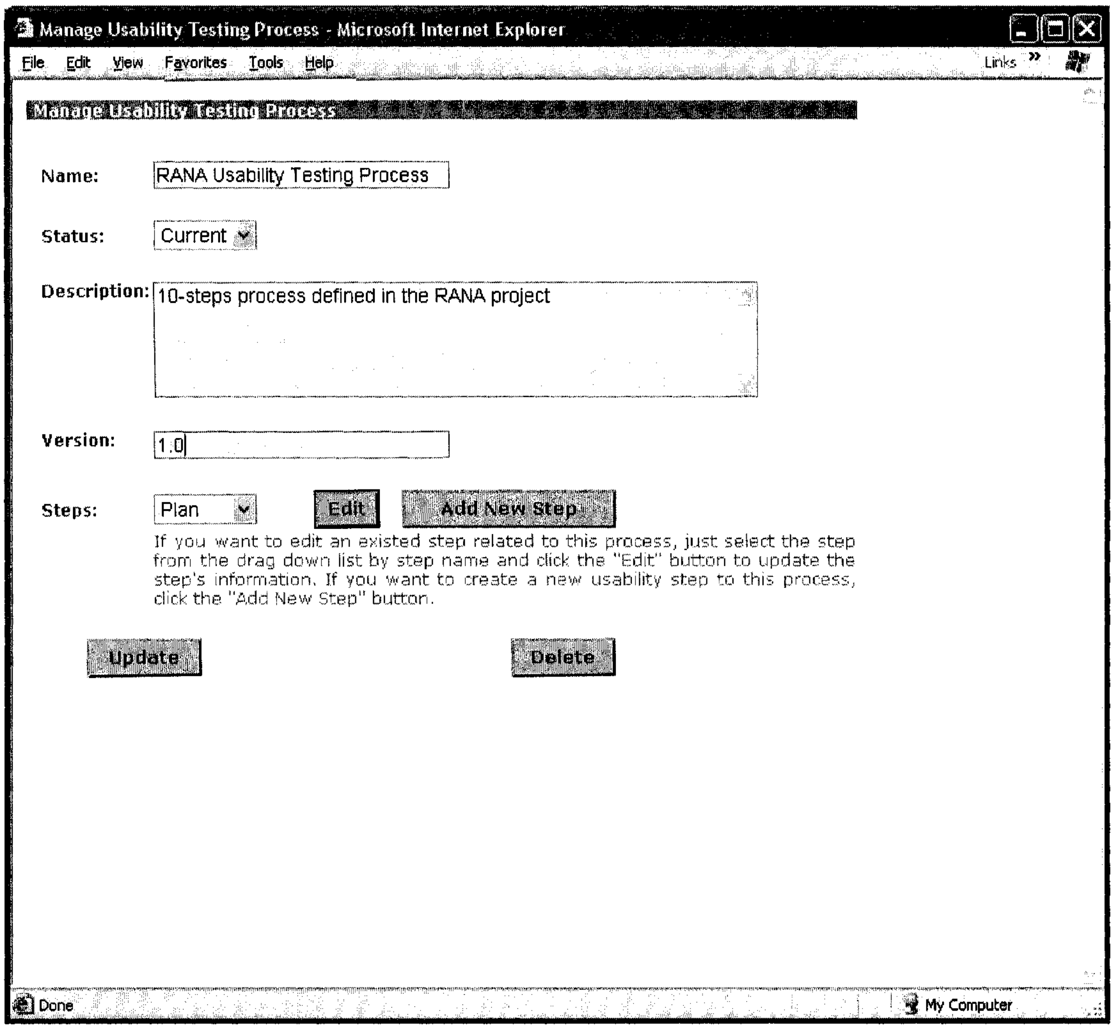


Figure 4-11: User Interface of Process Management

4.8.3. Usability Testing Process Visualization

The main tasks for usability testing process visualization function are:

- Retrieving usability process and its components information from usability knowledge base.
- Creating visualization HTML files by combining HTML visualization template with process information retrieved from knowledge base.
- Showing and modifying usability testing process status.

Figure 4-12 illustrates the interface of acquire step visualization. A visualization interface is composed of three functional layers:

1. ***The left side layer:*** process visualization section, identifies the process and its steps. The current step is always highlighted and some visual elements, green check and red cross help distinguishing between the steps already completed and the ones left to do. Thus the tester is always aware at which step of the process he/she is.
2. ***The top-right side layer:*** step visualization section, gives a snapshot of the current step. The activities are clearly shown as well as the actors, the methods, the tools, the deliverables and the available templates of the step. More information can be accessed for each of those elements by clicking on the elements' hyperlink. This layer also provides means to navigate through the process steps and to keep track of the current step's status.
3. ***The bottom-right side layer:*** component visualization section, is used as the target of the hyperlinks in the top-right side layer. For example, if the user clicks on a tool or method name in the top-right side layer, the result of that request is displayed is this layer. By default, when a step is loaded, this frame shows the global description of the current step.

The advantage of having such a layered-structure is that the main elements of the process are always visible and users can easily understand the process structure. Besides above three functional layers, on the top of the interface, we provide the *Help* and *Logout* buttons in case users have any difficulties or do not want to work through the whole process.

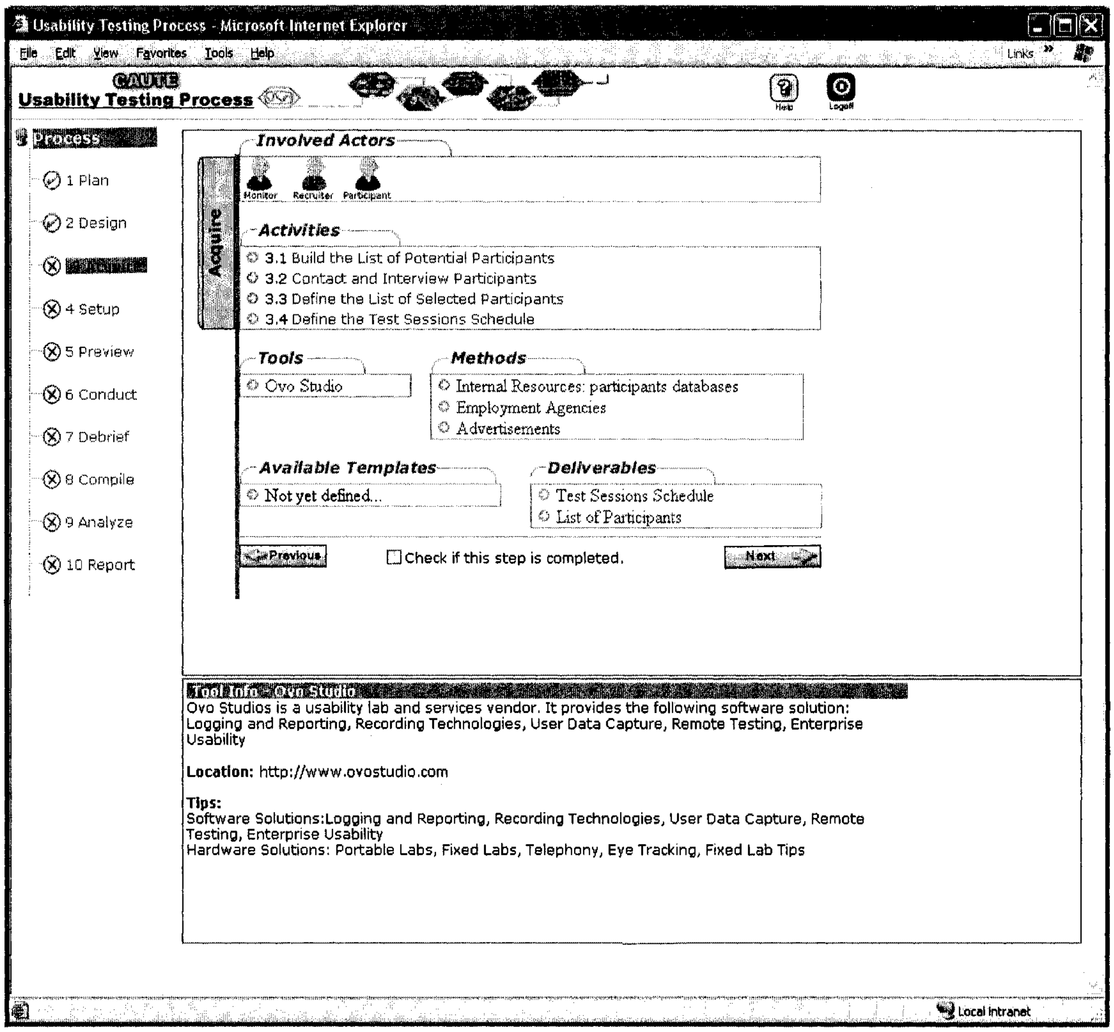


Figure 4-12: Visualization User Interface of Acquire Step

4.8.4. Usability Project and Test Management

Usability project can be defined as all pieces of work that are related to one or a set of products, such as software applications and websites that need to be tested. One project can contain more then one test, which associated to a specific goal. To be noticed, all functionalities addressed in this section can only performed by advance users.

Project Management Functionalities

- ***Create Usability Project:*** Usability Projects are identified uniquely by a project id and a project name and are composed of several Usability Tests (Figure 4-14). The project status at creation is set to "A" (Active).

- ***Lock Project:*** Allow to lock a Project (Set Project Status to "L"). All tests under the project are locked (not accessible until the project is unlocked). No further modifications are allowed to existing Tests and no new Test can be created until the Project is unlocked (Reset Project Status to "A").
- ***Unlock Project:*** Allow to unlock a previously locked Project. All Tests under project will be unlocked (Reset Project Status to "A").
- ***Close Project:*** Allow to permanently lock a Project (Set Project Status to "C"). Any Test under the Project will be only readable.
- ***Delete Project:*** Allow to physically delete a Project. All Tests under the project will be deleted.

Test Management functionalities

- ***Create Usability Test:*** A Usability Test is always created under a Project. A Project ID and a Test ID identify tests uniquely. A Manager is assigned to the test during the test creation process (by default the Manager is the person who creates the Test). A test is associated to one or more products. The Test status at creation is set to "A" (Active).
- ***Lock Test:*** Allow to lock a Test (Set Test Status to "L"). No further modifications are allowed until the test is unlocked.
- ***Unlock Test:*** Allow to unlock a previously locked Test (Reset Test Status to "A").
- **Close Test**: Allow to permanently lock a Test (Set Test Status to "C"). The Test will only be readable by a limited set of users to be specified according to the privileges, such as managers and evaluators.
- **Delete Test**: Allow to physically delete Test.

4.8.5. Data Collecting and Retrieving

Web forms are used to collect and retrieve data. For data collecting (Figure 4-13), on the left section of interface, we use tree structure to show the usability testing process. Users can switch between different steps by clicking the step title. Inside each step, there are several activities that need to be done. The right section is the web form where users enter the data. One web form corresponds to one or several activities on the left section. For example, in Figure 4-13, the web form contains the information related to the activities including define test objective and define target audience. After entering required information in the web form, users can click the navigation button that is located at the end of the web form to next or previous web form. At the same time, information that the user has entered will be automatically recorded into database.

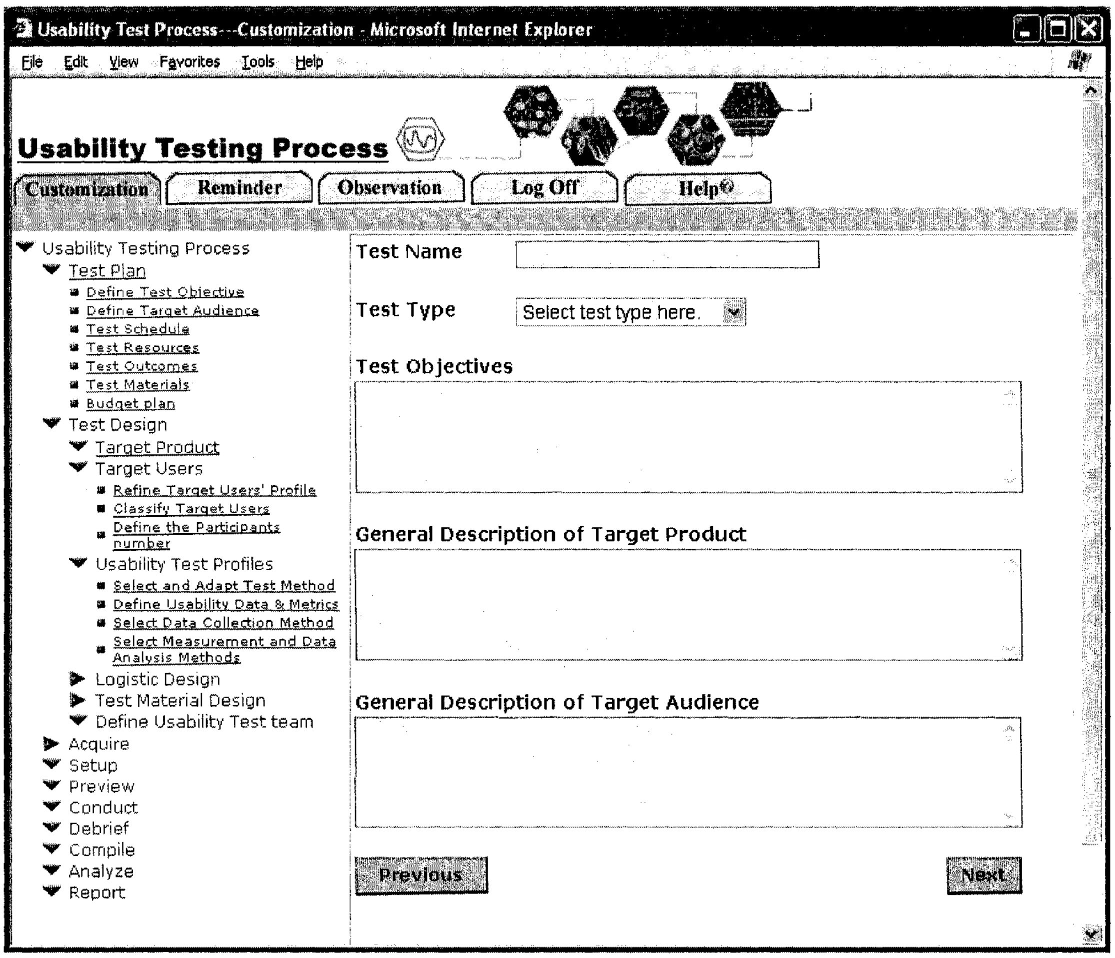


Figure 4-13: Data Collecting User Interface of Defining a Test

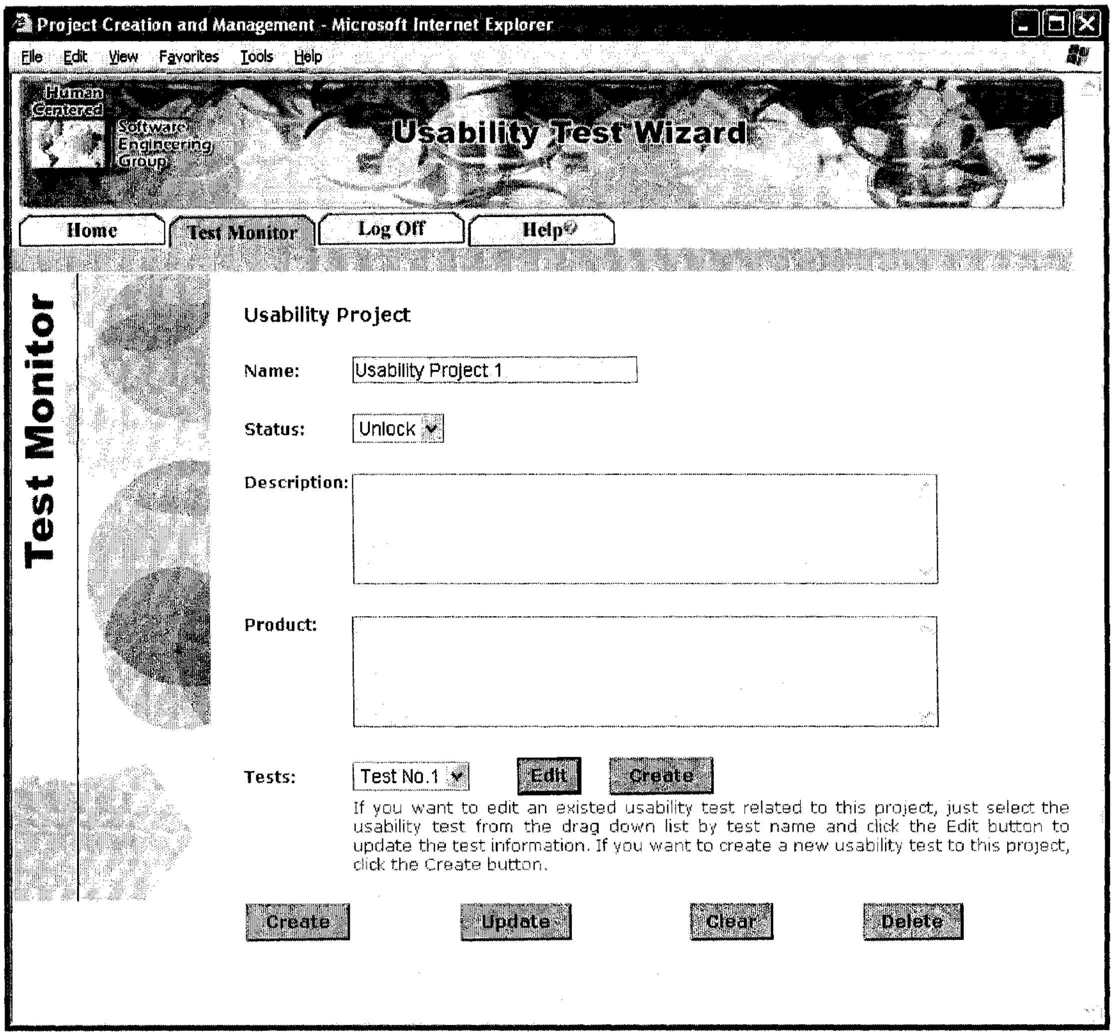


Figure 4-14: Data Retrieving User Interface of Usability Project

When users want to retrieve the data that has been stored in the database, firstly, they go to the corresponding data retrieving interface. Figure 4-14 shows an example of data retrieving interfaces. Secondly, users can enter the name of usability project or test that they want to retrieve, and then click update button below. Finally, WizUse searches the database according to the parameters that the user entered. If the data exists, then WizUse retrieves the data from the database and writes them into corresponding web forms. Otherwise, an error message will be sent out.

4.8.6. Report generation

When the user has entered all required information, WizUse can generate several documents for the user, such as usability test plan and usability testing report according to predefined format. In Appendix B, I provide our CIF usability report format. At first, documents are offered by HTML files. Later, PDF files will be also available.

4.9. Summary

In this chapter, first we gave some background introduction about RANA project and remote usability architecture since WizUse is a core component tool of RANA platform and the research related to this thesis is a subset of RANA project. Second, we presented some basic background information about "wizard", and explained why we chosen wizard as our solution for WizUse. And then, we introduced the design and implementation of WizUse. Finally, the major functionalities are presented. In a word, WizUse is to support in bridging the gap between usability testing processes and tools, and assist usability professionals to conduct usability tests.

5. Validation

5.1. An Online Survey on Usability Testing Processes and Practices

5.1.1. Objectives of the Survey

After having defined the proposed usability testing process model, we conducted an online survey on usability testing process and practices. This survey has two major objectives:

- ***Validate the proposed usability testing process.*** In our usability testing process model, we divided the whole process into ten steps, from *Plan* to *Report*. Along with these steps, we combined four key elements, actors, activities, methods & tools and outputs & artifacts. Although we consider all these components are important and necessary, they need to be validated from the industry where actual usability practices are being carried out. After this validation stage, the process model can become more practical, accurate and improved.
- ***Identify and collect from practitioners' current usability practices.*** When I did the literature reviews, most research work and documentation are in the time range from 1990 to 2000. In the recent years, the usability of software application and websites has become a hot theme both in academic research and industry daily agenda. New efforts to identify and collect the most recently usability progresses are extremely required. By doing so, we can know what practical usability testing process companies are using, how many companies have conducted remote usability testing, facilities, methods and the tools that are currently used, as well as the best practices and experience.

It needs to be noticed that defining the usability testing process model is an iterative procedure including several evaluation and modification iterations. The process model that is presented in the chapter 3 is the final version after we got feedback from our evaluations including this survey. Some modifications have been made to improve and refine the model.

5.1.2. Methodology

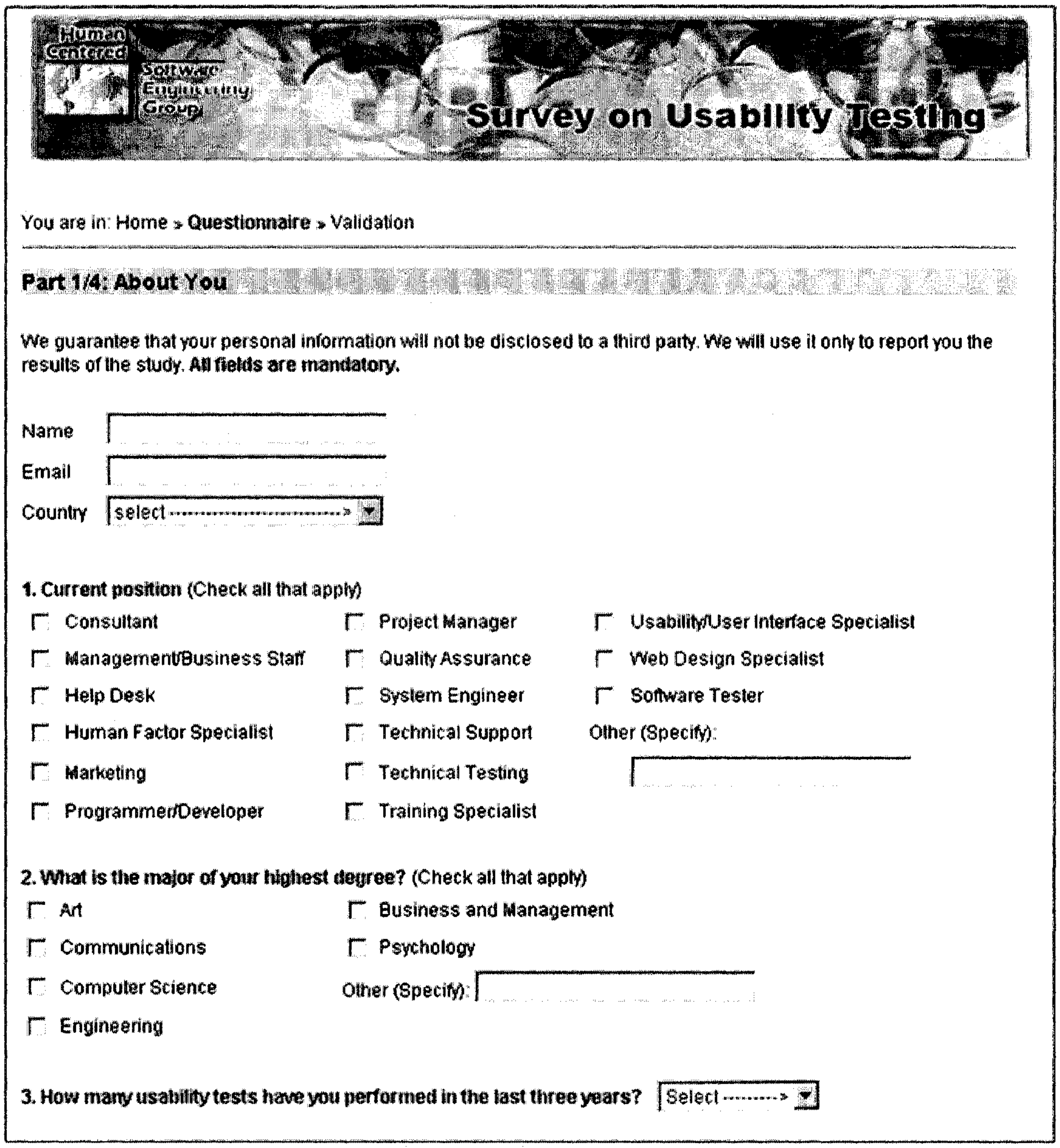
The Questionnaire

An online questionnaire is used to conduct our survey, which was divided into 4 sections of totally 12 questions. Two first sections were attempted to get information related to the respondents and their companies, for example the respondent's position and usability expertise. The third section was designed to gather the information of testing methods and practices. The last section is related to the infrastructure and tools used during the usability tests. At the end of the questionnaire, an open question was asked, in which respondents could address whatever they want. This open question enabled us to get comments on what is required to improve the usability tests in terms of activities, methods and tools.

According to some pre-tests we did ourselves, the questionnaire could be completed in between 10 and 15 minutes. The short completion time of the questionnaire was one of the strategies used to get as many respondents as possible. Moreover, we ensure our respondents that the information collected through this questionnaire will be strictly confidential and used only for fundamental research ends. All the data collected were stored in a MySQL database for further automatic analysis. Figure 5-1 is a screen shot of this online questionnaire. The whole version of the questionnaire is provided in Appendix C.

The Target Population

The target population of this survey was primarily composed of usability professionals who have an experience in conducting usability tests and researchers related in this domain. The questionnaire link was sent by email to some mailing lists and newsgroups related to usability including ACM-SIGCHI (Special Interest Group on Computer-Human Interaction), Usability Professionals' association (UPA) Montreal, UPA Toronto, Francophone Human-Computer Interaction Association (AFIHM), and Ergonomie des Interfaces Homme-Machine (ERGOIHM).



Survey on Usability Testing

You are in: Home » **Questionnaire** » Validation

Part 1/4: About You

We guarantee that your personal information will not be disclosed to a third party. We will use it only to report you the results of the study. **All fields are mandatory.**

Name

Email

Country select

1. Current position (Check all that apply)

- Consultant
- Management/Business Staff
- Help Desk
- Human Factor Specialist
- Marketing
- Programmer/Developer
- Project Manager
- Quality Assurance
- System Engineer
- Technical Support
- Technical Testing
- Training Specialist
- Usability/User Interface Specialist
- Web Design Specialist
- Software Tester
- Other (Specify):

2. What is the major of your highest degree? (Check all that apply)

- Art
- Communications
- Computer Science
- Engineering
- Business and Management
- Psychology
- Other (Specify):

3. How many usability tests have you performed in the last three years? Select

Figure 5-1: Screen Shot of Online Survey

Data Validation

In order to avoid partial filling out of the questionnaire, a JavaScript validation program is used. In case some questions are not answered, this validation program would inform the respondent to fill the questions accordingly, and the questionnaire could not be submitted. The validation system also prevented some biased data such as a wrong email, a string instead a number, etc. Before undertaking any detailed analysis, we went over all the responses in order to detect some inconsistency or incompleteness. Moreover, the questionnaire was made up of questions with multiple choices and lists of selection that avoid getting some inconsistent or inadmissible data. No missing values or inconsistent values were detected except that we detected two copies of the same questionnaire. We guess that this person submitted the form two times. The design of our survey and the pilot surveys considerably reduced the chances of getting incomplete questionnaires.

5.1.3. Results and Quantitative Data Analysis

We received 86 effective feedback within three months of conducting. After collecting the data, we used the statistic software, SPSS, to conduct analysis. The respondents mainly come from Canada (24.4%), United-States (32.6%) and France (34.9%), and this because the mailing-lists and newsgroups used were dedicated for North-American and French people (Figure 5-2).

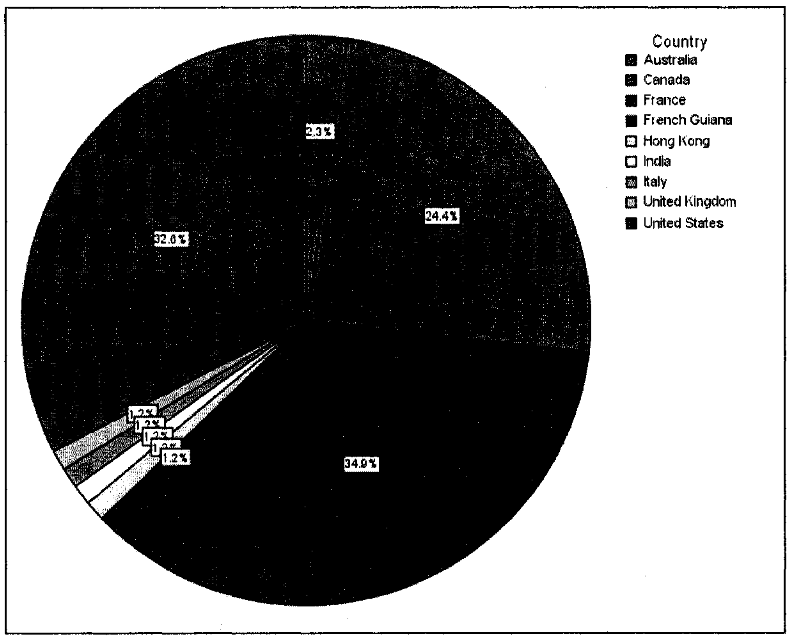


Figure 5-2: Country Distribution of Respondents

Respondents Profiles

The first section of the survey is designed to gather respondents' profiles. We asked four questions to let the survey participants provide their positions, major of the highest degree, the number of the tests that they conducted in the last three years and the level of expertise on usability.

The 86 respondents come from different business sectors (Table 5-1). 57 of them are usability/user interface specialist, 25 of them are human factor specialist, and 14 respondents are web design specialist.

Position Categories	Frequency
Consultant	36
Management/Business Staff	7
Human Factor Specialist	25
Marketing	4
Programmer/Developer	9
Project Management	9
Quality Assurance	2
System Engineer	1
Technical Support	2
Technical Testing	1
Training Specialist	4
Usability/User Interface Specialist	57
Web Design Specialist	14
Software Tester	4
Other	20

Table 5-1: Position Distribution of Respondents

As the analysis in target population of the survey, the respondents are mainly usability professionals who have an experience in conducting usability tests and related researchers in this domain. In order to distinguish the level of respondents, we used two criteria, the number of tests conducted and the expertise level that participant provided, to classify them into three different groups: novice, intermediate and expert (Figure 5-3) [Moha et al. 2005a]. After classifying, there are 8 experts, 66 intermediate professionals and 12 novices.

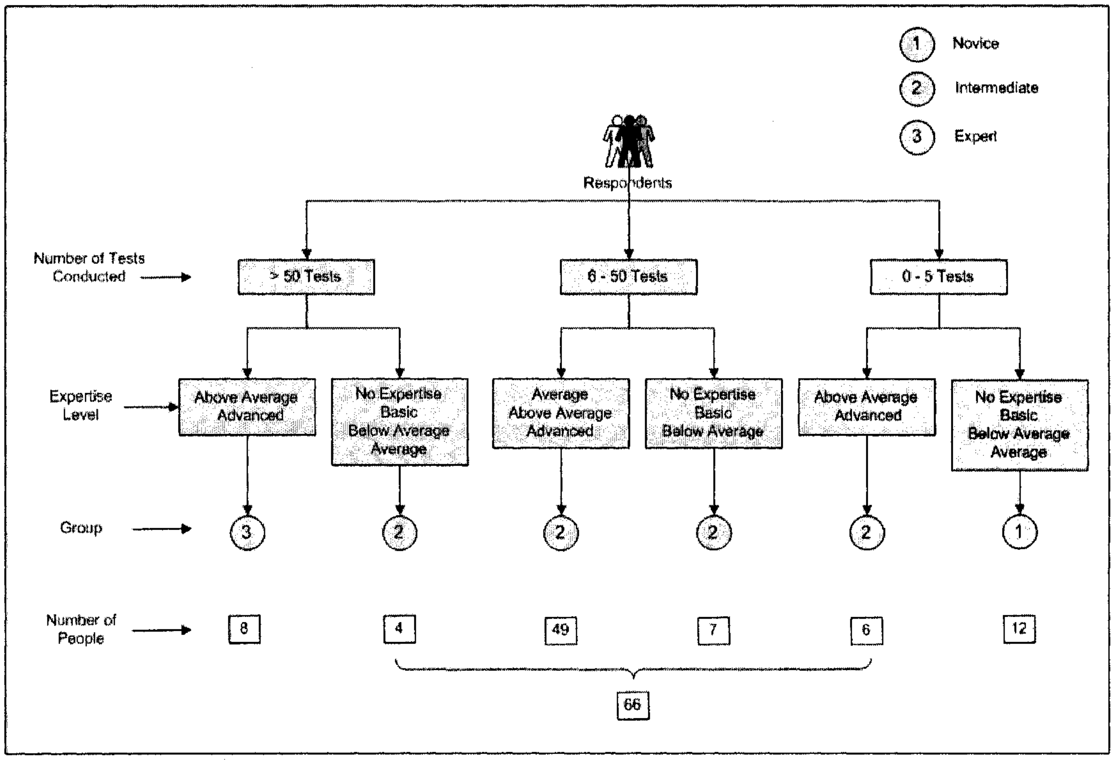


Figure 5-3: Classification of Respondents

Companies Profiles

The following table 5-2 gives the proportion in frequency and percentage of the companies that were categorized by the employee numbers. 24.4% respondents come from the companies that have more than 5,000 employees. 12.8% respondents come from the companies that have 1,001 to 5,000 employees while 27.9% respondents are either solo practitioners or coming from the companies whose sizes are less than 10 people. [Moha et al. 2005a]

33.7% companies have 3 to 5 years of experience in usability testing (Figure 5-4), and few companies have more than 15 years of experience (totally 9.4%).

Number of Employees	Frequency	Percentage
Solo practitioner	10	11.6
2-10 employees	14	16.3
11-20 employees	5	5.8
21-50 employees	4	4.7

51-100 employees	5	5.8
101-250 employees	2	2.3
251-500 employees	6	7.0
501-1,000 employees	5	5.8
1,001-5,000 employees	11	12.8
More than 5,000 employees	21	24.4
I don't know	3	3.5
Total	86	100.0

Table 5-2: Company Size Distribution

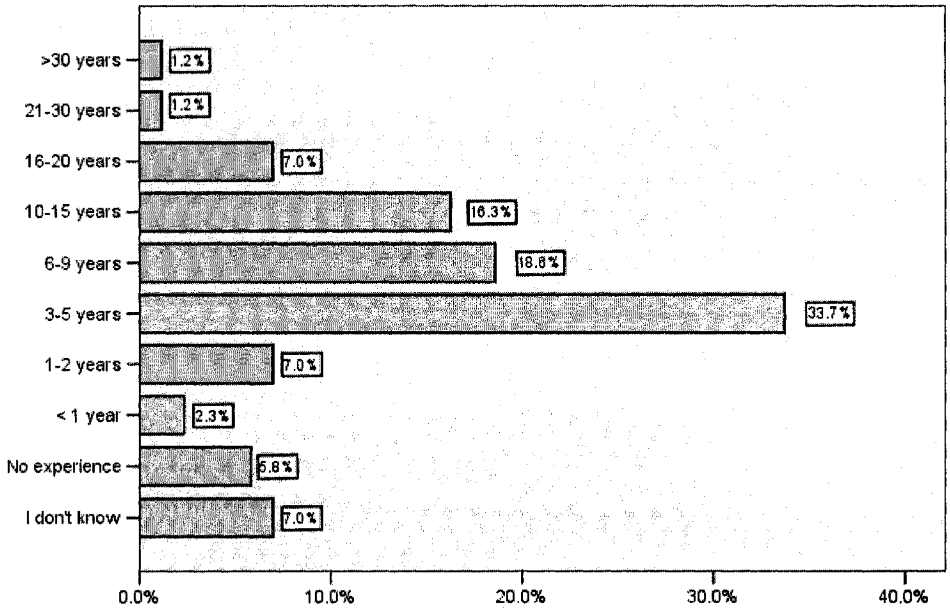


Figure 5-4: Experience Years of Company in Usability Testing

We also used cross-tab analysis to the relationship between different questions in our survey. Table 5-3 shows the relationship between company size and their experience in usability testing. According to the table, we found that large companies, more than 1,000 employees, normally have more experience compared with small size companies.

Company Size	Experience
Large (More than 1,000 employees)	≥ 6 years
Intermediate (100<employees<1,000	3-5 years
Small (<100 employees)	3-5 years

Table 5-3: Company Size vs. Experience

Activities, Methods and Materials of Usability Testing

In the last two section of our survey, we asked several questions related to the activities, methods and materials of conducting usability tests. Survey participants answer these questions by clicking the radio boxes that we predefined to evaluate the importance levels, including none, low, medium, high and very high.

Similarly, cross-tab analysis method is used to analyze the relationship between the importance levels and expertise levels of respondents (three groups classified above). In most of the cases, we got the similar distribution of importance level in three groups. However, we got different distributions in some cases. For example, the Figure 5-5 illustrates the analysis result of evaluating the importance of the activities "Select video demonstration clips". According to the figure, the majority of experts deem the importance level of the activity is low while most of intermediate respondents consider it is medium. When there is some diversity between different respondent groups, we give different weight for each group. The group that has more expertise has larger weight value. It means that experts' opinions are more important for us.

Table 5-4, 5-5, and 5-6 give the overall importance level of each usability method, activity, material based on our analysis [Moha et al. 2005a]. Some of the activities, methods and materials we put in our survey questions are ones that we are not sure for their relevancy related to usability tests. Questionnaire feedback gave us the points of view from the usability professionals' perspectives in industry. For example, according to respondents, the importance of the activity, select video demonstration clips, is low, so we refined our process model by changing this activity from recommended activity to an optional one.

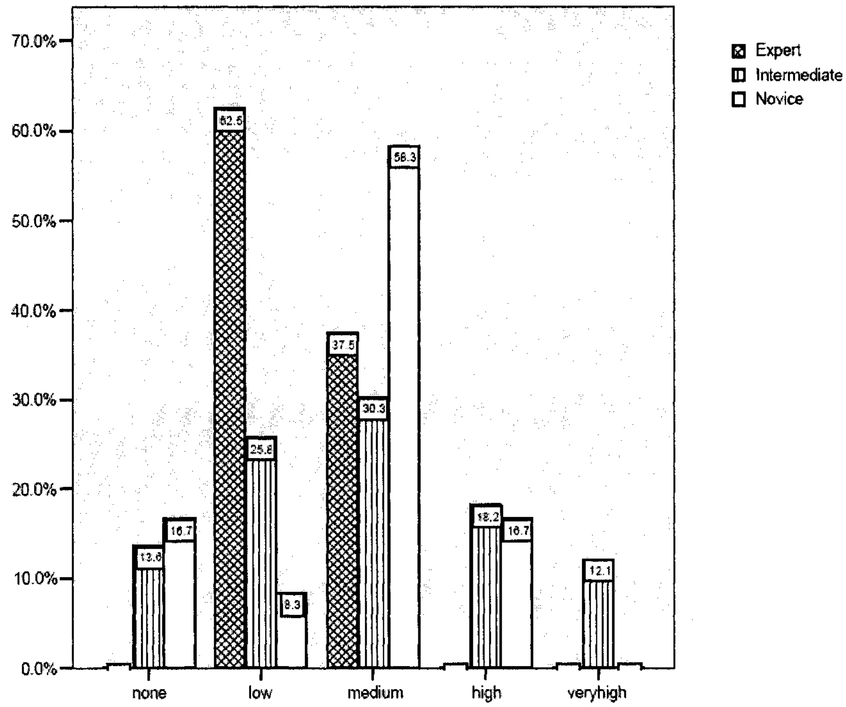


Figure 5-5: Test Activity: Select Video Demonstration Clips

Activity	Level
Define user profiles	Very high
Define the usability test team and tester profiles	Medium
Conduct a pre-test yourself	Medium
Conduct a pilot test	Very high
Collect informal comments from the participant after the test	Medium
Review the pre-test & post-test questionnaires with the participant	Medium
Add comments of usability testers and observers	High
Identify patterns and findings from the test results	Very high
Prepare a preliminary report	High
Select video demonstration clips	Low
Produce or update design guidelines	High

Table 5-4: Importance Level of Usability Testing Activities

Method	Level
Contextual inquiry Type of interview for gathering field data as much data as possible from users.	High
Critical incident technique End users are asked to identify specific incidents that they experienced personally.	High
Ethnographic study Observing users in the field.	High
Interview Query users about their experiences and preferences with the product.	Very high
Prototyping Model of the final product for testing some parts of the products.	Very high
Questionnaire Written lists of questions distributed to users.	Medium
Scenarios of use Scenarios specify how users carry out their tasks in a specified context.	Very high
Storyboard A storyboard is a low fidelity prototype consisting of a series of screen sketches.	Medium
Thinking aloud Encourage participants to say aloud what they are thinking.	High
Walkthrough A scenario-based experience of the product from the customers' perspectives.	High
Wizard of Oz Unimplemented technology to be evaluated by using a human to simulate the response of a system.	High
Focus group Informal assembly of users whose opinions are requested about a specific topic.	Low

Table 5-5: Importance Level of Usability Testing Methods

Material	Level
Screening questionnaire	High
Background questionnaire	High
Pre-test questionnaire	High
Post-test questionnaire	High
Test schedule	High
Non disclosure agreement	Medium
Tape consent form	High
Check list of test environment	High
Data collection forms	High
Preliminary report	Medium
Final report	High

Table 5-6: Importance Level of Usability Testing Materials

Usability Testing Tools

One of the questions in section 4 is related to the tools that are currently being used in usability testing. We listed the tools that were chosen from our tools survey as multiple choices and let respondents select the ones they used during their usability practices. We also gave an opportunity to the survey participants to provide their own tools that are not listed in the question. Table 5-7 presents the results after we analyzed the feedback from 86 respondents. In the table, I use bold letters to emphasize the mostly used tools. The first eight tools are listed in the question, and the others are the ones that respondents specified.

Tool	Frequency
Camtasia studio	**16**
Clementine data mining	0
Ergolight usability software	1
Sigma Stat, Sigma Plot SPSS product	**8**
Survey system	4
IBM Ezsort & Ezcalc	**9**
Noldus Observer	3

Ovologger	3
Bailey's Testing	1
Digital Video Camera	2
Camlan	1
Own logging software	**5**
Excel (data analysis)	**3**
Free software	1
HTML Prototype and video	1
Morae	**8**
Kronos	1
Leximancer	1
Macshapa	1
Mobile wireless UT lab	1
MSN Encoder	1
Own ad hoc software tools	1
RapidPlus	1
Self-developed tool	**2**
Usability Datalogger (see TZ)	1
Video recording	2
Virage	1
WebEx for remote testing	1
Word	1
Mobile wired UT Lab	1
Take notes by hands	1
Specific software patches	1
WebSurveyor	1

Table 5-7: Usability Testing Tools

According to the results, currently used usability tools mostly are mainly focusing on data collecting and analyzing. Some companies prefer developing their own tools to support usability tests. After this survey, for the highly used tools, we did full investigations and used them in our usability tests to verify their efficiency and effectiveness.

5.1.4. Qualitative Data Analysis

At the end of the survey, we provided an open question that let us collect respondents' comments. This qualitative data analysis enables us to review a certain number of important points related to the formalization of usability testing processes, usability tools development, and promotion of usability in industry.

The respondents highlighted the effort initiated by the usability community for standardization of testing procedures and usability testing reports. The CIF (Common Industry Format) is one example. This common industry format for reporting usability testing results became an ANSI standard in December 2001, and has become an international standard in May 2005. UPA is actually working on creating the knowledge base for usability professionals that would gather all testing practices.

The same phenomena occured in the domain of the user centered design. Indeed, most of the companies have developed common terminologies, models and tools for the data exchange between different professions of UCD [Venturi and Troost 2004]. However, this domain suffers from the lack of criteria, clear and accurate measures for estimating the importance and impact of their contribution in the industry of world [Vredenburg et al. 2002]. All this effort of standardization and formalization are a challenge for the usability community to ensure their development and acceptability within the companies.

A lot of respondents express the needs to have tools that enable them to not only collect and analyze test data, but also annotate and index videos automatically and extract video sequences from the huge quantity of video information. Morae, the only tool on the market, meets this requirement partly. Some respondents also express the need of having tools for data analysis that are not only easy to use and configure, but also reliable and affordable.

According to the collected comments, the value of usability testing has gained more and more recognition in industry. Company managers are convinced by the utility of such test. As a consequence, they are more willing to invest their money to purchase usability tools as well as infrastructures, and mobilize their staffs to conduct usability tests. This provides a nice atmosphere to formalize usability testing and develop usability tools.

5.1.5. Conclusion

This survey allows us to obtain the state of the art of actual practices for conducting usability testing in industry, and validate our process model. According to the analysis of feedback, we get the following points:

- What are the test activities, methods and materials that we need to consider when conducting usability tests.
- Even if the fixed usability lab based tests and onsite tests are remaining predominant in terms of test conducting environment, remote usability testing offers a new test platform and is becoming more and more popular.
- There is great demand for usability tools, especially the ones that can collect and analyze test data, and annotate and index video sequences.
- Usability is yet a young domain that is not well-established within companies, and is gaining more and more recognition.

Other research directions have been identified after this survey. The first research axis is to design and develop on the assistant tools for collecting and analyzing test data, and the tools for annotating and indexing video sequences. The second research axis is to propose a platform to conduct usability tests including the usability testing process, usability tools and methods.

We benefited a lot from the results of this survey to validate and refine usability testing process that we defined. However, it is important to specify that because of the number limitation of the questions, the survey does not deal with all the problems related to usability test and other evaluation methods, but gathers general information accordingly.

5.2. Expert Evaluation

During the period of WizUse design and implementation, we invited three usability experts to evaluate the interfaces as well as data modeling. After several iterations and evaluation cycles, we modified some parts of our design according to experts' recommendations. For instance, in our early design version, the process visualization interface is like the image A shown in Figure 5-6. After we got the feedback from the expert evaluation, we made four major modifications in the step visualization section and create our final version (the image B in Figure 5-6).

1. Integrate the components into different boxes with tabs on the top and show all detailed information in the component visualization section. By doing so, the page length is reduced, and thus prevents to use scrolling bar. Visualizing components into categories makes things easier for the users to create their mental model of the process structure.
2. Instead of using green, brown and red to depict privilege level, we let the system do the job, and make user privilege system transparent to users. In step visualization section of every step, the interface only shows the actors who have the privilege to access this step. Different colors are used to distinguish different actors.
3. Change the step title from the top to the side to fully use the width of the window.
4. Move the navigation bar from the top to the bottom.

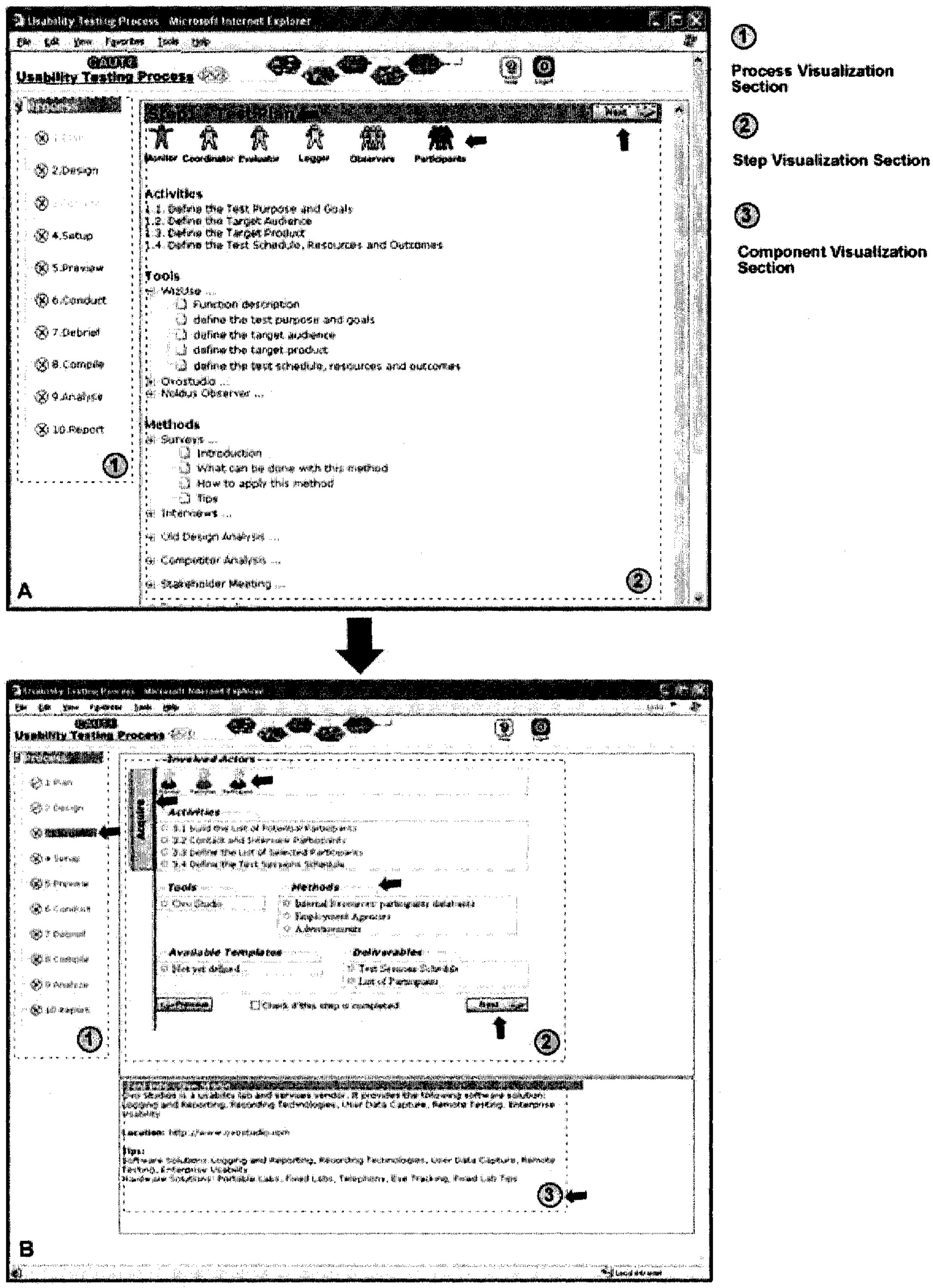


Figure 5-6: Visualization User Interfaces Before and After Expert Evaluation

5.3. Case Study: Osmose Evaluation

Between February and May 2005 we, Concordia's Human-Centered Software Engineering group, got a project to conduct a usability test of a Head Mounted Display (HMD) system, named Osmose. This project gave me a chance to validate WizUSe by using it to support our usability testing.

Osmose website defines the system as follows: "Osmose (1995) is an immersive interactive virtual-reality environment installation with 3 dimensional computer graphics and interactive 3D sound, a head-mounted display and real-time motion tracking based on breathing and balance." [Davies 1995] Osmose system is composed of the following components:

- A server (Figure 5-7), made up of two racks containing four Personal Computers (PCs): two with the Windows operating system, one with Macintosh and one with Linux. The racks also contain sophisticated multimedia processing hardware that provides a variety of audio-visual outputs in various formats.
- A vest (Figure 5-8), which when worn by a participant, measures his breathing (by measuring chest expansion via a pressure gauge) and the orientation of his upper body.
- A helmet (Figure 5-8), containing two low-resolution optical lens offering the user a 3D view, speakers providing surrounding background sounds, and a magnetic sensor that detects the helmet's position and orientation.

All the components are connected with each other by audio and video cables except the magnetic sensor and receiver, which are using wireless method to communicate.

During the usability tests, we mainly used process management and visualization functionalities of WizUse to support our testing. Firstly, we used process management function to customize the usability testing process, which was proposed in the chapter 3, based on the particularity of HMD project. Secondly, after the customization, we visualized the testing process by WizUse. Finally, our usability test team conducted the tests by following the testing process.

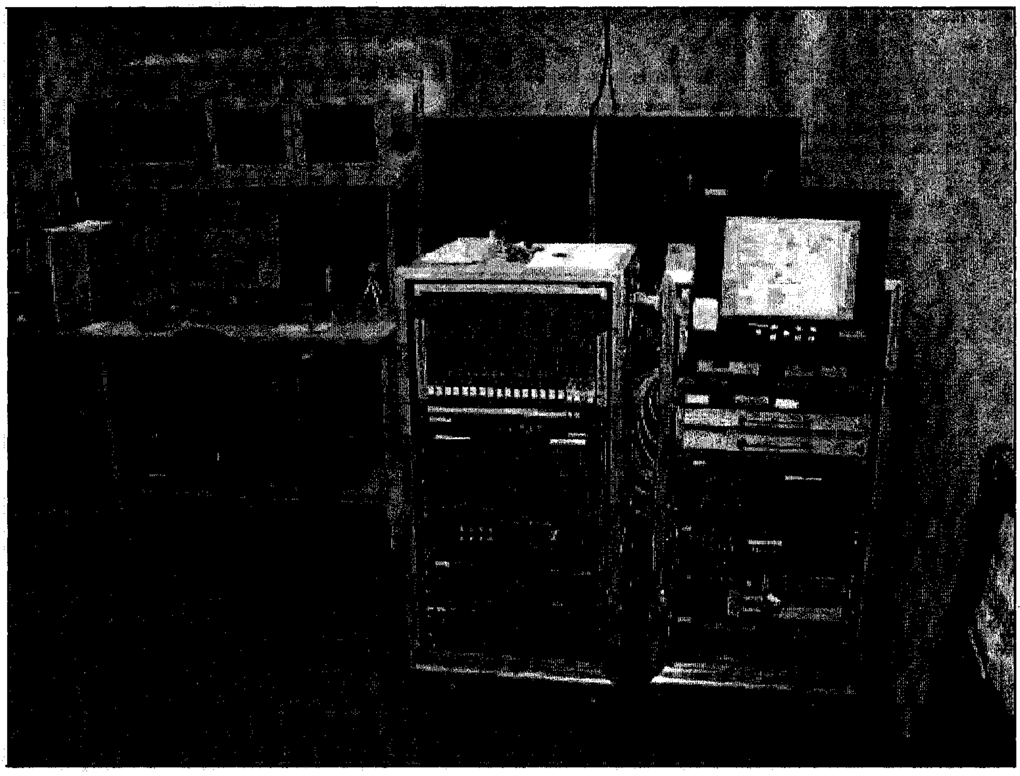


Figure 5-7: Osmose Server and Test Observation Area

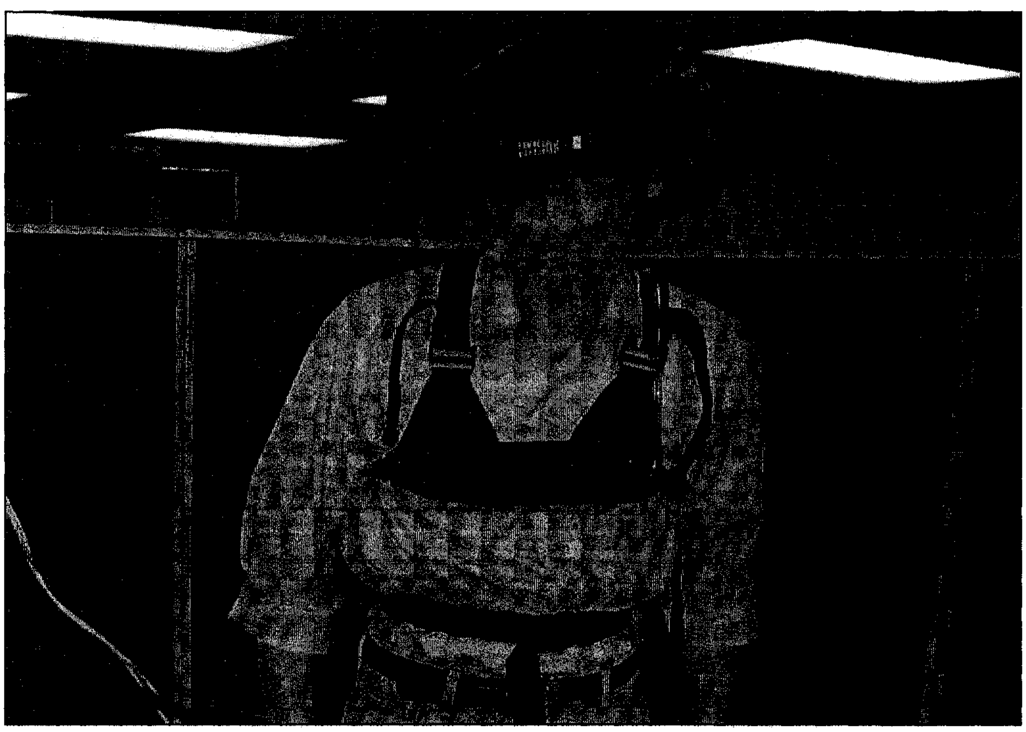


Figure 5-8: Helmet and Vest of Osmose System

We also referenced WizUse initial usability knowledge base for methods, tools and templates when we selected testing methods and tools as well as generated the test materials. For example, we chose thinking aloud protocol from usability method list in the conduct step of WizUse when we conducted the tests. Meanwhile, we applied Morae that was recommended by WizUse to automate the test conducting and data collecting.

In our process customization stage, we created four new actors and changed the definition of monitor as follows.

- ***Technician:*** The technician's original role was to create the usability testing environment for Osmose. Once testing began, his task changed to a support role. At the start of every testing day, the technician booted up Osmose and setup the testing environment. During the day's tests the technician helped calibrate the equipment to each participant, started and stopped recordings of Morae Recorder, and was generally available in case of technical issues. At the end of every testing day, the technician shutdown Osmose and securely stored the equipment.
- ***Welcomer:*** The welcomer assisted the participant before the usability test. She organized a meeting place and time with participants, greeted them and then gave them the consent form and the pre-test questionnaire. The pre-test questionnaire was answered by the participant without any help from the welcomer. Once this was complete, the welcomer introduced the participant to the monitor and her job was done.
- ***Data Logger:*** There were two types of data loggers: a manual data logger and an automated data logger. The manual logger wrote notes by hand, marking down when participants needed assistance, noting problems they had had or obstacles they had run into, and making general observations. The automated data logger recorded events using Morae Remote Viewer, marking off whether or not tasks were completed successfully, and marking when tasks were started and stopped for the purpose of timing them.
- ***Interviewer:*** The interviewer assisted the participant after the usability test. The interviewer gave participants the post-test questionnaire, which asked for information regarding their perspective on where they went in the virtual world,

what they did, whether or not they found the experience immersive, and how they would better implement the system were they the designer. The participants were not given a paper to fill out; the interviewer asked all of the questions and wrote down all of the answers. The reason for that was to leave the participant free to talk about the experience and help prompt the participant for useful information.

- ***Monitor:*** The monitor assisted the participant during the usability test. The monitor answered questions, and sometimes offered directions. He or she provided help for the participant in putting on and taking off the Osmose helmet, vest and wireless microphone. In conjunction with the technician, the monitor calibrated the breathing for the vest. He or she enforced the thinking aloud protocol, requiring participants to speak about what they were thinking and doing. Once the test was ready to begin the monitor turned off the room's lights, turned on the floodlight, and signaled the start of the test for the purpose of timing it. During the test the monitor made sure that the participant did not wander away from the magnetic sensor controller (the participant position in Figure 5-16). Once the test was complete, the monitor introduced the participant to the interviewer.

Moreover, we divided our test participants into four groups: general public, artists, kids and usability engineers.

Figure 5-9 demonstrates the Osmose testing environment that was designed by our technician. The testing room was divided into four functional areas.

- ***Reception Area.*** In this area, the welcomer greets participants and gives them the consent form and pre-test questionnaire to fill.

- ***Testing Area.*** During a test session, participant stands in the participant position, wearing the helmet and the vest, to interact with Osmose system. The manual data logger sits aside to make the notes by hand, and a camcorder was used to capture the participant's behaviors. This area is separated from other areas by cubicle partitions.

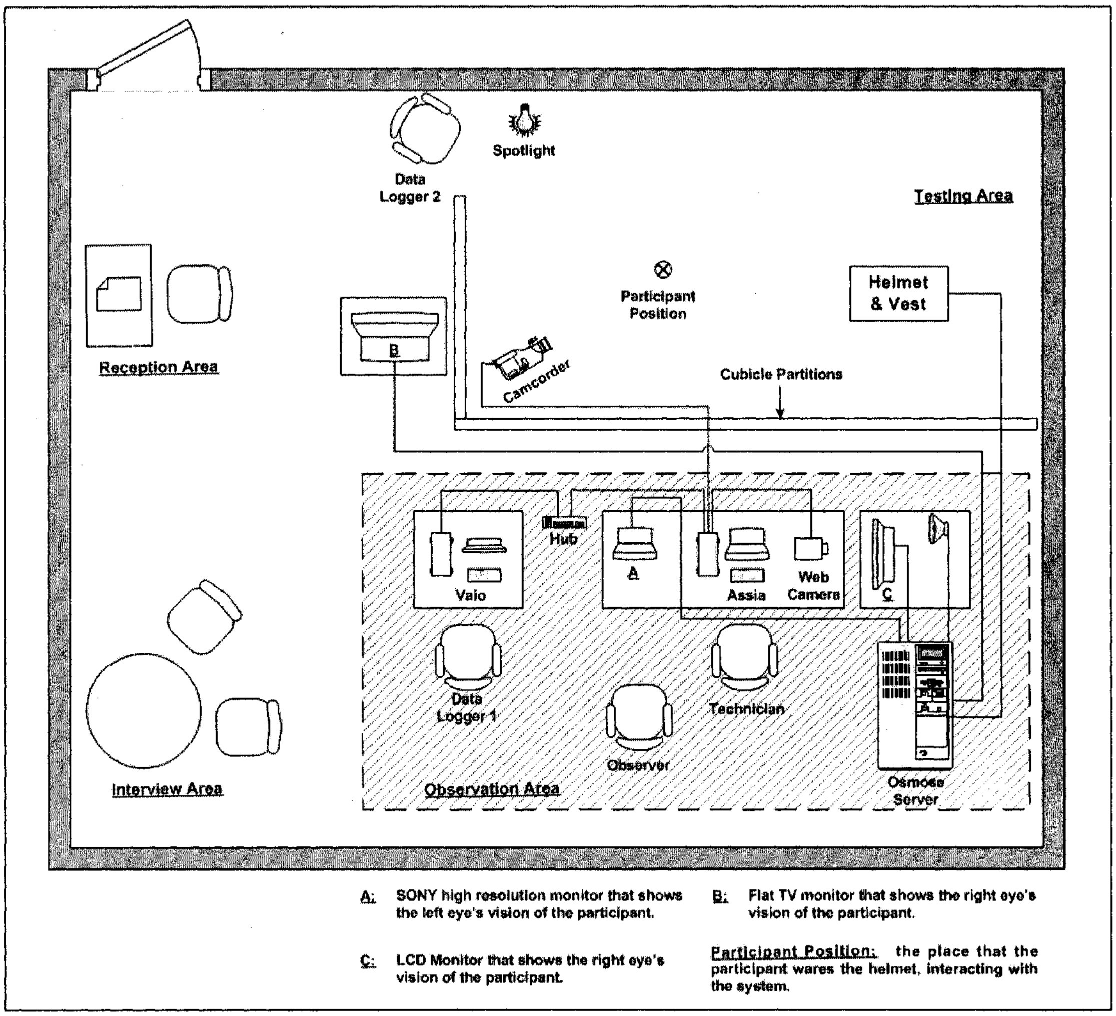


Figure 5-9: Osmose Testing Environment

- ***Interview Area.*** This is the place that the interviewer sits with the participants to review the test session. The participant is asked to fill a post-test questionnaire and give his/her comments.

- ***Observation Area.*** In this area, there are two desktops, Vaio and Assia, connecting with each other by a hub. (Assia has installed Morae Recorder to record participant's behaviors and his/her sight view under the control of the technician.) Morae Remote Viewer is installed in Vaio, where the automated data logger creates testing markers and records test events. Observers sit behind the data logger and the technician to observe the test. The web camera is used to shoot the LCD screen for capturing the participant's sight view and integrating it with the Morae's recording.

After process customization and test design, we followed the instruction from WizUse, and distributed tasks according to the actors. 30 test sessions were conducted within 6 weeks. HMD project is still ongoing, and now, we are in the usability analysis step.

5.4. Summary of Validation Results

Through expert evaluations, we improved our user interface design and database structure, thus increasing the usability of WizUse itself. The usability tests of Osmose system, gave us a chance to use the wizard in the actual usability tests and evaluate WizUse's efficiency and productivity. At the same time, we also had the opportunity to validate our proposed usability testing process.

After the validation, we found WizUse presented in this thesis is useful and offers a few advantages. Although it is not a tool that fully automates the whole testing process, it provides solid assistance and helps in conducting usability tests. WizUse is a kind of tutorial to usability testing process that will link to or suggest appropriate tools, standard reporting templates and methods at any given step of the process. Moreover, WizUse gives usability testers a clear answer about the question: *who is doing what, when, and how.* It guides and assists the usability professional throughout the testing process, and reduces the workload of testers.

6. Conclusions and Future Investigations

6.1. Summary of Contributions

In this thesis, I reviewed the major existing usability testing processes, and then proposed a new usability testing process. I then specified the WizUse, a web-based tool designed for supporting usability testing. Finally, I presented the validation efforts related to both proposed usability testing process model and WizUse. More precisely, my contributions of this thesis are as follows:

1. I co-defined the proposed usability testing process.
2. I designed the remote usability testing architecture of the RANA project and created a prototype.
3. I conducted a usability tools survey. Within the tools we reviewed, I selected the most useful ones for further investigation. Later, I generated a tools survey report, in which I gave a general description, advantages and disadvantages for each tool selected.
4. I co-conducted a survey on usability testing practices, and distributed the questionnaire to the usability professionals. After we collected all the feedback, I analyzed the data by using the statistical software, SPSS.
5. I designed the user interfaces of WizUse, and co-designed the database structure and functionalities.
6. I designed the usability testing environment and worked as a technician on the HMD project.

6.2. Conclusions

6.2.1. Proposed Usability Testing Process

Although the importance of usability testing is wildly recognized, as highlighted in our work, most of the existing HCI/usability testing processes are incomplete and informal. Most of them are just a description of individual working experiences in conducting the usability tests. Under this circumstance, a new usability testing process has been proposed in this thesis, which combined the major strengths and overcome the drawbacks of the current processes. The proposed ten-step process includes a detailed description of the activities, the actors, the usability methods and tools as well as the artifacts produced and used at each step. This detailed description gives a clear answer to the following question that is addressed in process modelling and makes a process more formal: who is doing what, when, and how throughout the usability tests.

We maintained the common steps, activities as well as templates and methods we selected form the current usability testing processes. All of this is the basis of our proposed process. Moreover, proposed process also combined the expertise and experiences we collected from the two surveys. The drawbacks that have been overcome in the new process model are as follows:

- Several milestones, which are missing in the current processes, were added, including hiring participants, compiling usability data and conducting pre-tests.
- Although some usability testing processes are defined several actors, these definitions are too general and need to be decomposed. In our proposed process, nine different kinds of actors as well as their roles were clearly defined and assigned to steps and activities.
- The usability tools that we selected form the tools survey were blended with steps and activities where they can be used. Thus, the gap between processes and tools was bridged. In addition, usability methods were distributed into steps, so usability professionals can know exactly when to employ them.

6.2.2. WizUse

As discussed in this thesis, there is a big gap between the existing processes and tools. To support in bridging this gap, the proposed process has been embedded into the web-based wizard, WizUse that provides assistance to usability professionals in conducting usability tests. WizUse visualizes usability testing process in a structured matter by combining steps, activities, usability tools, templates, methods, and actors together. All components can be accessed through a unique interface via Internet. Some of the advantages of WizUse as a process-assisted usability-testing platform are:

- Bridging the current gap between usability practices and tools
- Helping novice as well as experienced usability professionals;
- Providing step-by-step advice that can help in training staffs;
- Supporting usability testing performance (e.g. collecting usability data from web-based forms);
- Helping in customizing the process to different types of applications and empirical studies (e.g. users can design their own usability testing process by using the process management functionality.);
- Visualizing the usability testing process in a structured manner; and
- Facilitating the integration of some usability tools.

WizUse and proposed usability testing process were all validated by both expert evaluations and a series of usability tests that were conducted in the context of HMD project.

6.3. Future Investigations

Since we just finished the first stage of the WizUse implementation, there are still some issues that need to be further investigated.

First, our proposed usability testing process is designed for general purpose. It should be customized to meet the requirements of different types of usability testing, such as remote usability testing and fixed lab testing. In addition, since web-based applications are becoming more and more popular, the evaluation demands on web-based applications and web sites are continuously growing. A customized process for this kind of testing is really needed. Integrating these customized process packages into WizUse is extremely useful for the novice usability testers to follow.

Second, it is time to conduct real-world remote usability testing to validate our proposed process and the WizUse tool. Although we have some validation by doing expert evaluations and local usability tests, it is not enough. In a real remote environment, there are some external variables (such as management-related and technical) that are not concerns in a local testing environment.

Third, we should integrate QUIM into WizUse to provide usability measurement functionalities. QUIM, a framework for quantifying usability metrics in software quality models, provides a multi-layered hierarchical model that distinguishes five levels called factors, criteria, metrics, data, and artifacts (Figure A-2 in Appendix A). After integration, WizUse will offer usability professionals the usability metrics, criteria and factors that they can apply to collect usability data accordingly.

Finally, as I mentioned before, the second implementation stage of WizUse is the data level tools integration. In order to achieve this goal, an intermediate data structure is needed to exchange the data among different usability tools. XML language is a good choice. By using the intermediate data presentation, WizUse can interact with other usability tools we selected.

References

ANSI/INCITS 354-2001: Common Industry Format for Usability Test Reports, InterNational Committee for Information Technology Standards (INCITS).

C&C Usability Testing Process [Internet]. Available from <http://staff.washington.edu/cr/diagrams/utesting.htm#scoped> Computing & Communications UCD Group of University of Washington [Accessed February, 2005].

List of Methods [Internet]. Available from <http://www.usabilitynet.org/tools/list.htm> UsabilityNet [Accessed March, 2004].

Usability Software: OvoLogger [Internet]. Available from <http://www.ovostudios.com/ovologger.asp> Ovo Studios [Accessed February, 2004].

ISO/IEC 9126, Software product evaluation - Quality characteristics and guidelines for their use, 1991. Geneva, International Organization for Standardization.

ISO 9241-11, Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability, 1998. Geneva, International Organization for Standardization.

ISO 13407, Human-centred design processes for interactive systems, 1999. Geneva, International Organization for Standardization.

ISO TR 18529, Ergonomics - Ergonomics of human-system interaction - Human-centred lifecycle process descriptions, 2000. Geneva, International Organization for Standardization.

ISO/IEC 9126-1, Software engineering - Product quality - Part 1: Quality model, 2001. Geneva, International Organization for Standardization.

ABRAN, A., KHELIFI, A., SURYN, W., RILLING, J. & SEFFAH, A. (2003) Consolidating the ISO Usability Models. *In the Proceedings of FASE 2003 - Conference on Fundamental Approaches to Software Engineering*, 11(4), pp.8, Warsaw, Poland.

ALUCID Solution *Portable Labs* [Internet]. Available from <http://www.usabilitysystems.com/prod_usability_portlab.html> [Accessed May, 2005].

AMSDEN, J. (2001). *Levels of Integration: Five Ways You Can Integrate with the Eclipse Platform*. Eclipse Corner Article. March, 2001.

BARTEK, V. & CHEATHAM, D. (2003). *Experience remote usability testing, Part 1: Examine study results on the benefits and downside of remote usability testing* [Internet]. Available from <http://www-106.ibm.com/developerworks/library/wa-rmusts1/> [Accessed August, 2004].

BEVAN, N. (1999). Quality in Use: Meeting User Needs for Quality. *Journal of System and Software,* 49(1), pp. 89-96.

BEVAN, N. (2000). *Cost Benefit Analysis, TRUMP report* [Internet]. Available from <http://www.usability.serco.com/trump/methods/integration/costbenefits.htm> [Accessed November, 2003].

BEVAN, N. (2001). International Standards for HCI and Usability. *International Journal of Human Computer Studies,* 55(4), pp. 533-552.

BEVAN, N., CLARIDGE, N., MAGUIRE, M. & ATHOUSAKI, M. (2002) Specifying and evaluating usability requirements using the Common Industry Format: Four case studies. *In the Proceedings of IFIP 17th World Computer Congress,* pp.133-148, Montreal, Canada.

BEVAN, N. (2003). UsabilityNet Methods for User Centred Design. *In Human-Computer Interaction: Theory and Practice (Part 1), Volume 1 of the proceeding of HCI International 2003,* pp.434-438, Lawrence Erlbaum Associates.

BIAS, R. G. & MAYHEW, D. J. (1994). *Cost-Justifying Usability,* Boston, Morgan Kaufmann.

BOLLAERT, J. (2001). *Crafting a wizard: Fifteen dos and don'ts for designing wizards that make complex tasks easier for your users* [Internet]. Available from <http://www-128.ibm.com/developerworks/usability/library/us-wizard/> [Accessed September, 2004].

BOLLAERT, J. (2002). *Guidelines to help you develop and design your own wizard* [Internet]. Available from <http://www-128.ibm.com/developerworks/usability/library/us-wizard2/?article=wir> [Accessed September, 2004].

BORMANS, J., GELISSEN, J. & PERKIS, A. (2003) MPEG-21: The 21st century multimedia framework. *IEEE Signal Processing Magazine,* vol. 20, pp. 53-62.

DAVIES, C. (1995). *Osmose Description* [Internet]. Available from <http://www.immersence.com/osmose/osmose.htm> [Accessed March, 2005].

DILLON, A. (2001). The evaluation of software usability *In the 1st Edition of the International Encyclopedia of Ergonomics and Human Factors (2001),* London, Taylor and Francis.

DILLON, R. F. (2003). *Usability Testing: Myths, Misconceptions and Misuses* [Internet]. Available from <http://www.carleton.ca/hotlab/hottopics/Articles/usability_myths.html> [Accessed October, 2004]

DONAHUE, G. M. (2001) Usability and the Bottom Line. *IEEE Software,* Vol.18, No.1, January/February 2001, pp. 31-37.

DRAY, S. & SIEGEL, D. (2004) Remote possibilities? International usability testing at a distance. *ACM Interactions,* Volume11 pp.10-17.

DUMAS, J. S. & REDISH, J. C. (1999) A Practical Guide to Usability Testing, Revised Edition. Intellect, Ltd (UK).

EHRLICH, K., ROHN, J. (1994) "Cost Justification of Usability Engineering: A Vendor's Perspective." *In BIAS, R. G. & MAYHEW, D. J. (1994) Cost-Justifying Usability,* pp.73-110, Boston, Morgan Kaufmann.

GAFFAR, A., SINNIG, D., JAVAHERY, H. & SEFFAH, A. (2003) MOUDIL: A Comprehensive Framework for Disseminating and Sharing HCI Patterns. *In the Proceedings of CHI 2003 workshop on HCI Patterns: Concepts and Tools*, pp.3, Fort Lauderdale, United States.

GELLNER, M. & FORBRIG, P. (2001) Modeling the Usability Evaluation Process with the Perspective of Developing a Computer Aided Usability Evaluation (CAUE) System. *INTERACT 2001 Workshop on usability throughout the entire systems development lifecycle*, pp.8, Tokyo, Japan.

GOOD, M., SPINE, T. M., WHITESIDE, J., & GEORGE, P. (1986) User-derived impact analysis as a tool for usability engineering. *In the proceedings of Human Factors in Computing System: CHI '86, eds. MANTEI, M. & OBERTON, P.,* pp.241-246, New York, ACM Press.

HARTSON, H. R., CASTILLO, J. C., KELSO, J., KAMLER, J. & NEALE, W. C. (1996) Remote Evaluation: The Network as an Extension of the Usability Laboratory. *In the Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '96)*, pp.228-235, Vancouver, Canada.

HILBERT, D. M. & REDMILES, D. F. (1998) Agents for collecting application usage data over the Internet. *In the Proceedings of the second international conference on Autonomous agents*, pp.149-156, Minneapolis, Minnesota, United States.

HODGSON, P. (2003). *Usability Test Data* [Internet]. Available from <http://www.userfocus.co.uk/articles/datathink.html> [Accessed January, 2005].

HOLTZBLATT, K. & BEYER, H. (1993) Making customer-centered design work for teams. *Communications of the ACM,* 36(10), pp. 92-103.

IVORY, M. Y. & HEARST, M. A. (2001) The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys,* 33(4), pp. 470-516.

JEFFRIES, R. & DESURVIRE, H. (1992) Usability Testing vs. Heuristic Evaluation: Was there a Contest? *SIGCHI Bulletin,* 24(4), pp. 39-41.

JOKELA, T., LIVARI, N., MATERO, J. & KARUKKA, M. (2003) The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. *In the Proceedings of Latin American conference on Human-computer interaction*, Vol. 46, pp.53-60, Rio de Janeiro, Brazil

KANTNER, L. & ROSENBAUM, S. (1997) Usability Studies of WWW Sites: Heuristic Evaluation vs. Laboratory Testing. *In the Proceedings of the 15th annual international conference on Computer documentation*, pp.153-160, Salt Lake City, United States.

KARAT, C-M. (1994) "A Business Case Approach to Usability Cost Justification." *In BIAS, R. G. & MAYHEW, D. J. (1994) Cost-Justifying Usability,* pp.45-70, Boston, Morgan Kaufmann.

KING, M. (2003) Living up to standards. *In the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, Budapest, Hungary.

KRAUSS, F. S. H. (2003) Methodology for remote usability activities: A case study. *IBM SYSTEMS JOURNAL,* VOL 42(4), pp. 582-593.

KRUCHTEN, P. (2001) *The Rational Unified Process: An Introduction*, Addison Wesley.

LANDAUER, T. K. (1995) *The Trouble with Computers: Usefulness, Usability and Productivity*, MIT Press.

LAW, L. & HVANNBERG, E. T. (2002) Complementarity and Convergence of Heuristic Evaluation and Usability Test: A Case Study of UNIVERSAL Brokerage Platform. *In the Proceedings of the second Nordic conference on Human-computer interaction*, pp.71-80, Tampere, Finland.

LEFF, A. & RAYFIELD, J. T. (2001) Web-Application Development Using the Model/View/Controller Design Pattern. *In the Proceedings of the Fifth IEEE International Enterprise Distributed Object Computing Conference*, pp.118-127, Seattle, Washington, United States.

MAYHEW, D. J. MANTEI, M. (1994) "A Basic Framework for Cost-Justifying Usability Engineering." *In BIAS, R. G. & MAYHEW, D. J. (1994) Cost-Justifying Usability,* pp. 9-43, Boston, Morgan Kaufmann.

MAYHEW, D. J. (1999) *The Usability Engineering Lifecycle - A Practitioner's Handbook for User Interface Design,* San Diego, Morgan Kaufmann.

MOHA, N., LI, Q., GAFFAR, A. & SEFFAH, A. (2005a) *Enquête sur les pratiques de tests d'utilisabilité. In the Proceedings of IHM'05, 17ème Conférence Francophone sur l'Interaction Homme-Machine*, pp.115-122, Toulouse, France.

MOHA, N., LI, Q., SEFFAH, A. & MICHEL, G. (2004a) Towards a Platform for Usability Remote Tests via Internet. *In the Proceedings of OZCHI2004*, pp.22-24 Wollongong, Australia.

MOHA, N., LI, Q., SEFFAH, A. & MICHEL, G. (2004b) Analysis of Usability Labs for Remote Tests via Internet. Human Centered Software Engineering Group of Concordia University.

MOHA, N., SEFFAH, A., LI, Q. & NGANOU, A. (2005b) *Towards a Formal Modeling of the User-Oriented Testing Process.* Human Centered Software Engineering Group of Concordia.

NACK, F. & LINDSAY, A. T. (1999) Everything You Wanted to Know about MPEG-7: Part 1. *IEEE Multimedia,* vol. 6, pp. 65-77.

NAGHSIN, R., SEFFAH, A. & KLINE, R. (2003) Cognitive Walkthrough + Persona = An Empirical Infrastructure for Understanding Developers Experiences. *In the Proceedings of IEEE Symposia on Human-Centric Computing Languages and Environments*, Auckland, New Zealand.

NGANOU, A. (2005) RANA WizUse: A Web-Based Wizard of Usability Testing Process, *Computer Science.* Montreal, Concordia University.

NIELSEN, J. (1992) Finding usability problems through heuristic evaluation. *In the Proceedings of SIGCHI conference on Human factors in computing systems*, pp.373-380, Monterey, California, United States.

NIELSEN, J. (1994a) Guerrilla HCI: using discount usability engineering to penetrate the intimidation barrier. IN ACADEMIC PRESS, I. (Ed.) *Cost-justifying usability.* Orlando, FL.

NIELSEN, J. (1994b) *Usability Engineering,* San Francisco, Morgan Kaufmann.

NIELSEN, J. & MOLICH, R. (1990) Heuristic evaluation of user interfaces. *In the Proceedings of SIGCHI conference on Human factors in computing systems: Empowering people*, pp.249-256, Seattle, Washington, United States.

OLSEN, H. (2002). *The Bottom-line of Prototyping and Usability Testing* [Internet]. Available from <http://www.guuui.com/issues/02_02.php> [Accessed June, 2004].

PADDA, H. R. (2003) QUIM Map: A Repository for Usability/Quality in Use Measurement, *Computer Science.* Montreal, Concordia University.

PADDA, H. R., SEFFAH, A. & STRIKA, J. (2003) QUIM: A Tool and Knowledge Map for Usability Measurement. *In the Proceedings of the 13th International Workshop on Software Measurement (IWSM 2003)*, pp.93-112, Montreal, Canada.

RAJANEN, M. (2003) Usability Cost-Benefit Models – Different Approaches to Usability Benefit Analysis. *In proceedings of 26th Information System Research Seminar in Scandinavia (IRIS26),* Haikko, Finland.

ROSSON, M. B., CARROLL, J. M. (2002) *Usability Engineering: Scenario-Based Development of Human-Computer Interaction,* San Francisco, Morgan Kaufmann.

RUBIN, J. (1994) *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, John Wiley & Sons.

SCANLON, T. (1997). *When to Develop a Wizard* [Internet]. Available from <http://www.uie.com/articles/wizard/> User Interface Engineering [Accessed October, 2004].

SEAMAN, C. B., MENDONCA, M. G., BASILI, V. R. & MIM, K. Y. (2003) User Interface Evaluation and Empirically-Based Evolution of a Prototype Experience Management Tool. *IEEE Transactions on Software Engineering,* 29, pp. 838-850.

SEFFAH, A., KECECI, N. & DONYAEE, M. (2001) QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models. *In the Proceedings of Second Asia-Pacific Conference on Quality Software (APAQS'01)*, pp.311-318, Hong Kong, China.

SEFFAH, A. & METZKER, E. (2004a) The obstacles and myths of usability and software engineering. *Communications of the ACM,* 47(12), pp.71-76.

SEFFAH, A., MOHA, N. & LI, Q. (2004b). *RANA: Remote Architecture for Net-Based Analysis* [Internet]. Available from <http://rana.cs.concordia.ca> [Accessed May, 2004].

SHARMA, R., STEARNS, B. & NG, T. (2002) *J2EE Connector Architecture and Enterprise Application Integration, 1st edition*, Addison Wesley Longman, Inc.

SHULL, F., CARVER, J. & TRAVASSOS, G. H. (2001) An empirical methodology for introducing software processes. *In the Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, pp.288 - 296, Vienna, Austria.

SMIT, E. (1996) *Usable usability evaluation: If the montain won't come to Mohammed, Mohammed must go to the mountain,* London, Taylor & Francis.

SUN-MICROSYSTEM (2001) *Java Look and Feel Design Guidelines: Advanced Topics*, Addison-Wesley Professional.

TECHSMITH (2004). *Morae White Paper: An Overview of Morae* [Internet]. Available from <http://www.techsmith.com/products/morae/whitepaper.asp> TechSmith Corporation [Accessed December, 2004].

TULLIS, T., FLIESCHMAN, S., MCNULTY, M., CIANCHETTE, C. & BERGEL, M. (2002) An empirical comparison of lab and remote usability testing of web sites. *In the Proceedings of Usability Professionals' Association Conference*, Orlando, United States.

UPCHURCH, L., RUGG, G. & KITCHENHAM, B. (2001) Using Card Sorts to Elicit Web Page Quality Attributes. *IEEE Software,* 18, pp. 84-89.

VENTURI, G. & TROOST, J. (2004) Survey on the UCD integration in the industry. *In the Proceedings of the third Nordic conference on Human-computer interaction*, Vol. 82, pp.449-452, Tampere, Finland.

VREDENBURG, K., MAO, J. Y., SMITH, P. W. & CAREY, T. (2002) A survey of user-centered design practice. *In the Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pp.471-478, Minneapolis, Minnesota, USA.

WHARTON, C., RIEMAN, J., LEWIS, C. and POLSON, P. (1994) The Cognitive Walkthrough Method: A Practitioner's Guide. pp.105-140, *In J. Nielsen and R. Mack (eds.) Usability Inspection Methods,* New York, Wiley.

WICKHAM, D. P. (2001). *Link wizards and dialogs together to help users get through complex tasks* [Internet]. Available from <http://www-128.ibm.com/developerworks/usability/library/us-launch/> [Accessed June, 2004].

WIXON, D. & JONES, S. (1996). Usability for fun and profit: A case study of the design of DEC Rally Version 2. In M. Rudisell, C. Lewis, P.B. Polson, & T.D. McKay (Eds.), *Human-Computer Interface Design: Success Stories, Emerging Methods, Real-World Context* pp.3–35, San Francisco: Morgan Kaufmann.

ZHANG, Z., BASILI, V. R. & SHNEIDERMAN, B. (1999) Perspective-Based Usability Inspection: An Empirical Validation of Efficacy. *Empirical Software Engineering,* 4(1), pp. 43-69.

Appendix A. Tools Survey

1. Camtasia Studio

Camtasia studio provides an integrated environment to capture the screen activity and sounds of any MS Windows desktop. It also can edit and distribute the video files that captured by Camtasia Studio. Camtasia Studio can be divided into five parts.

- Camtasia Recorder — Records all desktop activity, including narration
- Camtasia Producer — Edits audio and video, and produce clips in various formats including RealMedia, Windows Media, Flash.
- Camtasia Effects — Enhanced recorded videos by Adding captions, text boxes and graphics.
- Camtasia Menu Maker — Indexes you media content for deliver videos.
- Camtasia Player

Advantages:

- Easy to record usability tests on software prototypes: Camtasia Studio records where they click and move. Since it also records audio, it captures the voices of the tester and the facilitator.
- It simple - Just press record and stop: It can takes about a minute to encode an hour of a test, and doing the output is also painless – in just a few steps.
- Quality: The audio remains good, and it's very easy to post it on a Web site.
- Inexpensive - $ 299
- Variety: Its support Real Player, and Macromedia Flash output.

Disadvantages:

- Does not record video of the testers.
- Live viewing has a delay of several seconds
- Viewers must install a codec.
- Viewers must download multi-Mb files (or test manager must edit video)

2. Survey System

The Survey System is software package available for working with telephone, online and printed questionnaires. It handles all phases of survey projects, from creating questionnaires through data entry, interviewing or email or Web page Internet surveys to producing tables, graphics and text reports. The Survey System was designed specifically for questionnaires.

Advantages:

- Cover most types of survey and questionnaire
- Manage surveys procedure at a professional level.
- Compare with traditional methodology, more efficient and more powerful.
- Easy to edit. Provide different views of collected data.

Disadvantages:

- Compare to other products, expensive.
- Separate tasks into different modules.
- A little bit complex, need some time to learn how to use it.

3. OvoLogger

OvoLogger Freeware is part of usability software solution provided by OVO Studios (Downloadable by request at www.ovostudios.com). It enables user to:

- Define scenario material
- Record scenario resolution and time
- Generate usability findings with associated logged observations
- Track development's implementation of usability recommendations
- Generate HTML-based usability report

It is more than just a simple usability test logger. It is designed to be used throughout the whole development life cycle, not just the testing phase.

Advantages:

- Easy to learn and use
- Efficient scenario resolution record technique

- Editable label, categories
- Useful observations review tool
- Great reporting tool
- Progress notes (follow up)

Disadvantages:

- Still too many Usability issues
    - License agreement to accept at each start
    - Report formatting, ask knowledge of HTML to user
    - No save button on the tool bar, an exit icon on exit bar
    - Log observations use can be confusing
    - Windows not resizable
- Project properties pane could be better "card sorted" everything is mixed; data collected before, during and after the test
- The label in the project properties are "editable" but it's not indicated
- User/Category: Only Add or Delete, no Modify... when trying to modify, it adds a new item
- Bad error prevention (e.g.: message "Run-time error '400': Form already displayed; can't show modally ")
- Lack of reliability: a fatal error encountered during evaluation
- Trying to too do much (too much half implemented features)

4. IBM Annotation Tool

IBM MPEG-7 Annotation Tool assists in annotating video sequences with MPEG-7 metadata.

Main functionality:

- Segment the Video sequence into shots automatically.
- Annotated shots in the video sequence.
- Customize description lexicon sets.
- Associated annotated descriptions with each video shot.
- Output annotated description as MPEG-7 descriptions in an XML file

Advantages:

- Segment video sequence into shots
- Classify frames into shots
- Associate annotations with shots, but store them separately
- Support temporal and spatial annotation
- Customize the annotation items
- Use MPEG7 extended XML to store annotations

Disadvantages:

- Video sequence decomposition: Video shot divided only by detecting the scene cuts, dissolves, and fades
- Insufficient video edition functionalities: Only supports mpg / mpeg video formats. For example, we can not switch a frame from one shot to another
- No enough error prevent mechanism: For example, when you want to save an MPEG 7 XML file which will overwrite an existed file

5. IBM Ezsort, Ezcalc

Ezsort & Ezcalc are Tools to help interface designers organize information based on users' expectations using statistical cluster analysis.

Advantages:

- It is Free
- It has advantages of computer applications (storing the result as files, reviewing...)
- It allows remote study (Not web)
- It generates Statistical probabilities of relationship between the card sorting results.
- It does not have a hassle of physical card sorting such as preparation and storage of the cards.
- It has reliable results.

Disadvantages:

- There is no place for user feedback or any survey.

- It is single user and do not have an online version. Therefore No email option especially for website analysis.
- No minor category naming.
- Only one level grouping
- You can't name the subgroups
- No persona consideration (no user category)
- No undo.
- They are not very visually appealing to users. Problems with automatic window resizing.
- No return to the previous step

6. Microsoft NetMeeting

NetMeeting lets users hold video conference calls, send text messages, collaborate on shared documents, and draw on an electronic whiteboard over the Internet or an intranet.

Advantages:

- IP-based video conferencing software: combine video and audio signals together, and synchronize them.
- Provide remote control feature: allow users to control remote computer.
- Provide several different level of sharing: file, software, and desktop level.
- Provide whiteboard: let users collaborate with each other.

Disadvantage

- When we use the remote control function, the visual quality of remote desktop in our local computer is not so high. The main problem is the colour.
- Network dependence: firewall and gateway can affect the usage of NetMeeting.
- Don't provide the function to record the video-audio signals and log files.

7. Advanced Digital Video Storage and On-line Retrieval System

ADViSOR Project: http://advisor-project.com

ADViSOR aims at boosting the state of the art in video annotation and retrieval technologies to the extent that formal information from videos (such as shot boundaries, camera movements and scene clusters) is extracted (semi) automatically. The technologies provided will free archivists from highly time-consuming mechanical tasks when searching for and extracting relevant semantic information from video and image.

ADViSOR provides an integrated system bringing together:

- Indexing processing: Shot boundary detection, automatic thumbnail extraction, scene clustering, camera movement analysis, object detection, object movement analysis, human face detection, speech analysis, keyword extraction optical character recognition, low-level thumbnail extraction.
- Classified manual annotation: Manual annotation are organized according to semantic criterion (who, when, what, where...) as proposed by MPEG7 multimedia description.
- Innovative search engine: It permits end users to combine at the same time
    - Query by keywords
    - Query on semantics video or image data, i.e. combined search based on textual annotation, motion characteristic, human face presence...
    - Result list sorting by visual image similarity

8. Clementine Data Mining Tool

Clementine is a data mining workbench that enables you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision-making. Clementine combines a wide range of visualization, analytical techniques together, using graphic interface to make system easy to use. This data mining tool is widely used in bioinformatics, marketing and decision-making of government contexts. Also it can be used in software usability evaluation area.

Advantages:

- Clementine supports full data mining process: data access, transformation, modeling, evaluation and deployment.
- Integrated wide range of visualization techniques to present huge amount of data, accelerate the understanding key relationships in a complex set of data, making it easy to find patterns behind of data.
- Clementine's powerful visual interface—combined with time-saving process support tools—enables users to use business expertise to quickly interact with your data and discover solutions in the shortest amount of time possible.

9. Sigma Stat, Sigma Plot SPSS Products

SigmaStat offers a full range of the statistical procedures used most often by scientists, including t-tests, analysis of variance, nonparametric statistics, correlation, linear and nonlinear regression, rates and proportions, normality and equalvariance testing.

Advantages:

- It provides full range of professional statistical tool for survey (in market, medical, and social survey).
- **SigmaStat's** "expert system" design intelligently guides the users through their entire statistical analysis to reduce the risk of making statistical mistakes and help them draw the right conclusions.
- Automatically find the right test with the Advisor Wizard, check if the users' data fit the underlying assumptions of their statistical model, effortlessly handle messy data and print a report complete with an explanation of the results.
- Strong graphical function;

Disadvantage:

- Lack embedded SQL function like SAS;
- Lack embedded function in cell like Excel,
- Compare with SAS, its data size is smaller,
- Compare with Excel, it is too professional for the non-statistical user.

SigmaPlot graphing software from SPSS Science takes users beyond simple spreadsheets to help them show off their work clearly and precisely. With SigmaPlot, users can produce high-quality graphs.

Advantages:

- Using effective graphic interface and the interactive Graph Wizard leads users through every step of graph creation.
- Provide more then 80 kinds of 2-D, 3-D technical graphic templates
- Users can customize their graphic in detail
- SigmaPlot offers seamless Microsoft Office integration, so users can easily access data from Microsoft Excel spreadsheets and present their results in Microsoft PowerPoint® presentations or output them into graphic programs.

10. Netopia Timbuktu

Netopia Timbuktu is a scaleable, multiplatform solution for user support, systems management, telecommuting, and collaboration across a LAN, WAN, the Internet or dial-up connections. By using this tool, systems administrators and network management personnel can control remote machines on the enterprise network to perform complete evaluations, diagnosis and troubleshooting.

Advantages:

- Provide bi-directional text/audio message exchange
- Users can switch between remote control interface and their own desktop
- Platform independence
- The tool can be used for usability testing for its capacity to control and observe remote computer activity

Disadvantages:

- It doesn't incorporate user observation.
- Don't provide the log file to record the user's interaction with the computer system

11. NetRracker ResearchManager Recorder

ResearchManager Recorder is a sophisticated, Java-based, remote-research capability that enables businesses to observe screen activity and interact with customers, partners and/or employees, empowering researchers to remotely view and record web and computer usage. The tool provides screen sharing, screen recording, real-time chat, white boarding, remote control and other capabilities that allow direct, real-time interaction with users during the testing process. Recorder allows users to be in their natural setting, for increased statistical validity. By grounding research results with direct observation of real customers, researchers are informed of page-level, moment-by-moment interaction problems, and the development team has concrete evidence to rely on when making project priority decisions.

12. QUIM Editor

(Adapted from [Padda 2003])

Quality in Use Integrated Measurement (QUIM) editor is a tool developed by the human-centered software engineering (HCSE) group to integrate different usability standards, quality models in one centralized knowledge base. In its latest version 2.0 (Figure B-1), it mainly offers two different kinds of functionalities, which are:

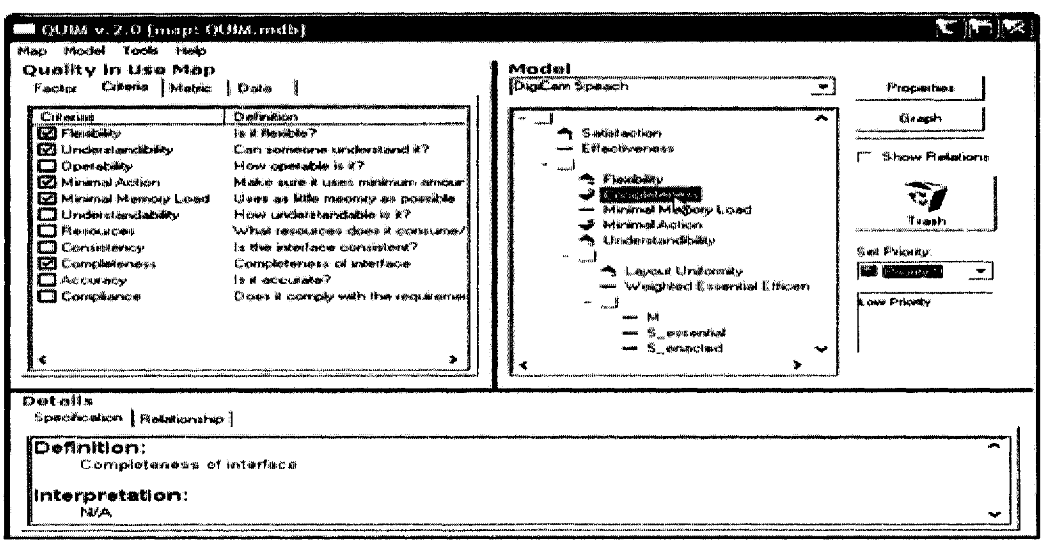


Figure A-1: Screen Shot of QUIM Editor

- Acting as a quality in use map, it enables an administrator to add, delete or modify the quality in use factors, criteria, metrics and data to the database or repository. In this way it offers an environment to measure the different aspects of quality in use.
- Acting as a model viewer, it allows the users to easily access any usability standard or quality model that can be presented as QUIM's interactive measurement model map. It also helps the users to create their own model based on the stored database of quality in use factors, criteria and metrics. It has the tendency to visualize the relationships that exist between quality in use factors, criteria, metrics and data. There is an added functionality of QUIM editor also, which is to prioritize the factors, criteria, metrics and data as well so that most cost effective and appropriate software characteristics that meet the user goals could be recognized and measured first.

Structure of QUIM

QUIM is a multi-layered hierarchical model like other software engineering models. It distinguishes five levels called factors, criteria, metrics, data, and artifacts (Figure B-2 [Seffah et al. 2001]). The relationship between these levels is an N-M relationship.

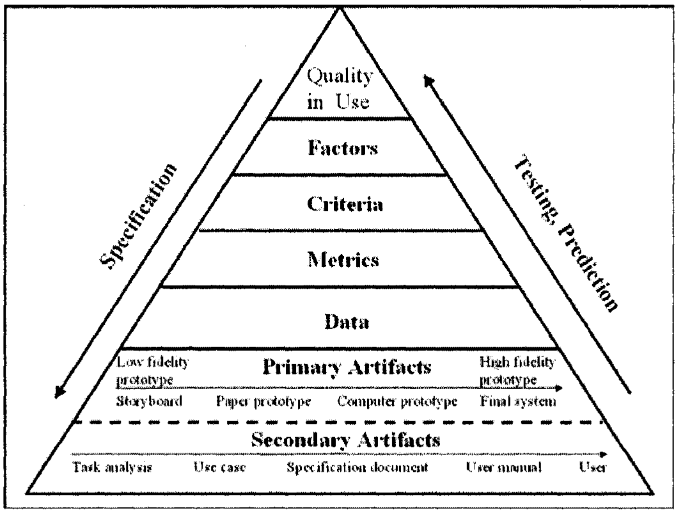


Figure A-2: The Hierarchy of QUIM

Here is the brief introduction of each of the term depicted in the structure.

Quality in Use

At the top of the hierarchy is the quality in use, which is perception or feeling of the end user about software quality. The user is mainly interested in looking at the external observable aspects of the software product like the speedy performance, good layout etc. The user is not interested in the internal attributes of a software product.

Factors

A quality in use factor is an attribute or characteristic of the user interface that the user sees in terms of the quality in use of the software product. The users define the quality of the software product in their own terms. It is not easy to measure and specify them. A Factor could be refined into the sub factors or Criteria. In QUIM, for the present study we have considered and investigated a set of 10 factors, they are: Efficiency, Effectiveness, Productivity, Satisfaction, Learnability, Safety, Trustfulness, Accessibility, Universality and Usefulness. They will be investigated in this thesis.

Criteria

Similar to other models a criterion is a sub factor or sub characteristic of the software product. Criteria are specified and defined in the language of software developers, so it is hard for most of the users to understand the technical terms involved while defining them. A criterion can be measured by more than one metric. In QUIM, at present there are a total of 27 criteria that can be measured through a number of metrics.

Metrics

As defined in previous chapter, it is a function whose output is a numeric value that summarizes the status of specific user interface characteristic. In QUIM, we have identified around 130 usability metrics, some of them are functions and described in terms of formula, others are just simple countable data.

Data

The lowest layer of QUIM is the list of data that is used to calculate metrics. We are concerned about two types of data, countable and calculable. The examples of both types of data are presented as follows:

- Countable

  For example: number of individual items on the screen, number of colors used or time spent while performing a particular task. The data collected from questionnaires also falls into this category.

- Calculable

  For example: A metric given by (Bevan, Macleod, 1994) for task effectiveness (TE) is:

  TE = Quantity × Quality / 100

  Where, Quantity is the percent of task completed and Quality is the percent of goal achieved.

  The percent of task completed and percent of goal achieved are the calculable data that are used for computing this metric.

13. MOUDIL

(Adapted from [Gaffar et al. 2003])

Montreal Online Usability Digital Library (MOUDIL) is an Integrated Pattern Environment (IPE) developed by Human Centered Software Engineering (HCSE) Group. MOUDIL was originally designed with two major objectives: Firstly, as a service to UI designers and software engineers for UI development. Secondly, as a research forum for understanding how patterns are really discovered, validated, used and perceived.

In particular, MOUDIL provides the following key features:

- MOUDIL has been designed to accept proposed or potential patterns in many different formats or notations. Therefore patterns in versatile formats can be submitted for reviewing.
- An international editorial board for reviewing and validating patterns. Before publishing, collected and contributed patterns must be accessed and acknowledged by the editorial committee. We are inviting HCI patterns practitioners and researchers to join this committee.

- Pattern Ontology editor captures our understanding of pattern concepts and puts them into relation with each other (Taxonomy).
- The MOUDIL Pattern Editor allows us to attach semantic information to the patterns. Based on this information and our ontology, patterns will be placed in relationships, grouped, categorized and displayed.
- The pattern navigator provides different ways to navigate through patterns or to locate a specific pattern. The pattern catalogue can be browsed by pattern groups or searched by keyword. Moreover, a pattern wizard will find particular patterns by questioning the user.
- The pattern viewer provides different views of the pattern, adjusted to the preferences of the specific user.

Appendix B. CIF Usability Report Template

[Enter Company Logo or Name; may be omitted]

[1,2]Common Industry Format for Usability Test Report v2.0

Comments and questions about this format: iusr@nist.gov

[❖ Name the product and version that was tested]

[❖ Who led the test]

[❖ When the test was conducted]

[❖ Date the report was prepared]

[❖ Who prepared the report]

For:

[❖ Customer Company Name]

[❖ Customer Company contact person]

Address inquiries to: [❖ Contact name(s) for questions and/or clarifications]

Phone: [❖ Enter supplier's phone number]

Email: [❖ Enter supplier's email address]

Address: [❖ Enter supplier's mailing or postal address]

[1] Legend for use: Items prefixed with a ❖ are Required; items prefixed with a ◆ are recommended.

[2] This template is Appendix C of the Common Industry Format v2.0 draft standard.

Executive Summary

[❖ Provide a high level overview of the test]

[❖ Name and describe the product]

[❖ Summary of method(s) including number and type of participants and tasks]

[❖ Results expressed as mean scores or other suitable measure of central tendency]

[◆ Reason for and nature of the test]

[◆ Tabular summary of performance results]

[◆ If differences are claimed, the associated statistical probability.]

Introduction

Full Product Description

[❖ Formal product name and release or version]

[❖ Describe what parts of the product were evaluated]

[❖ The user population for which the product is intended]

[◆ Any groups with special needs]

[◆ Brief description of the environment in which it should be used]

[◆ The type of user work that is supported by the product]

Test Objectives

[❖ State the objectives for the test and any areas of specific interest]

[❖ Functions and components with which the user directly and indirectly interacted]

[◆ Reason for focusing on a product subset]

Method

Participants

[❖ The total number of participants tested]

[❖ Segmentation of user groups tested, if more than one]

[❖ Key characteristics and capabilities of user group]

[❖ How participants were selected; whether they had the essential characteristics]

[❖ Differences between the participant sample and the user population]

[❖ Table of participants (row) and characteristics (columns)]

	Characteristic 1	**Characteristic 2**	**…**	**Characteristic N**
Participant 1				
Participant 2				
…				
Participant N				

[◆ Description of groups with special needs]

Context of Product Use in the Test

[❖Any known differences between the evaluated context and the expected context of use]

Tasks

[❖ Describe the task scenarios for testing]

[❖ Explain why these tasks were selected]

[❖ Describe the source of these tasks]

[❖ Include any task data given to the participants]

[❖ Completion or performance criteria established for each task]

Test Facility

[◆ Describe the setting, and type of space in which the evaluation was conducted]

[◆ Detail any relevant features or circumstances which could affect the results]

Participant's Computing Environment

[❖ Computer configuration, including model, OS version, required libraries or settings]

[❖ If used, browser name and version; relevant plug-in names and versions]

<u>Display Devices</u>

[❖ If screen-based, screen size, resolution, and color setting]

[❖ If print-based, the media size and print resolution]

[❖ If visual interface elements can vary in size, specify the size(s) used in the test]

<u>Audio Devices</u>

[◆ If used, specify relevant settings or values for the audio bits, volume, etc]

<u>Manual Input Devices</u>

[◆ If used, specify the make and model of devices used in the test]

Test Administrator Tools

[❖ If a questionnaire was used, describe or specify it here]

[◆ Describe any hardware or software used to control the test or to record data]

Experimental Design

[❖ Describe the logical design of the test]

[❖ Define independent variables and control variables].

[❖ Describe the measures for which data were recorded]

Procedure

[❖Operational definitions of measures]

[❖ Operational definitions of independent variables or control variables]

[❖Time limits on tasks]

[❖Policies and procedures for interaction between tester(s) and subjects]

[◆ Sequence of events from greeting the participants to dismissing them]

[◆ Non-disclosure agreements, form completion, warm-ups, pre-task training, and debriefing]

[◆ Verify that the participants knew and understood their rights as human subjects]

[◆ Specify steps followed to execute the test sessions and record data]

[◆ Number and roles of people who interacted with the participants during the test session]

[◆ Specify if other individuals were present in the test environment]

[◆ State whether participants were paid]

Participant General Instructions

[❖ Instructions given to the participants (here or in Appendix)]

[❖ Instructions on how participants were to interact with any other persons present]

Participant Task Instructions

[❖ Task instruction summary]

Usability Metrics

[❖ Metrics for effectiveness]

[❖ Metrics for efficiency]

[❖ Metrics for satisfaction]

Results

Data Analysis

[❖ Data Scoring]

[❖ Data Reduction]

[❖ Statistical Analysis]

Presentation of the Results

Performance Results

[❖ Tabular Performance Results per task or task group]

User #	Measure 1	Measure 2	...	Measure N
Participant 1				
Participant 2				
...				
Participant N				
Mean				
Standard Dev.				
Min				
Max				

[❖ Summary Table(s) of Performance Results across all tasks]

User #	Total Measure 1	Total Measure 2	...	Total Measure N
Participant 1				
Participant 2				
...				
Participant N				
Mean				
Standard Dev.				
Min				
Max				

[❖ Graphical Presentation of Performance Results]

Satisfaction Results

[❖ Tabular Satisfaction Results]

User #	Scale 1	Scale 2	…	Scale N
Participant 1				
Participant 2				
…				
Participant N				
Mean				
Standard Dev.				
Min				
Max				

[❖ Summary Table(s) of Satisfaction Results]

[❖ Graphical Presentation of Satisfaction Results]

Appendices

[❖ Custom Questionnaires, if used]

[❖ Participant General Instructions]

[❖ Participant Task Instructions]

[◆ Release Notes]

Appendix C. Survey on Usability Testing Processes and Practices

You are in: Home » Questionnaire » Validation

Part 1/4: About You

We guarantee that your personal information will not be disclosed to a third party. We will use it only to report you the results of the study. All fields are mandatory.

Name

Email

Country select ---------------------------->

1. Current position (Check all that apply)

- Consultant
- Management/Business Staff
- Help Desk
- Human Factor Specialist
- Marketing
- Programmer/Developer
- Project Manager
- Quality Assurance
- System Engineer
- Technical Support
- Technical Testing
- Training Specialist
- Usability/User Interface Specialist
- Web Design Specialist
- Software Tester

Other (Specify):

2. What is the major of your highest degree? (Check all that apply)

- Art
- Communications
- Computer Science
- Engineering
- Business and Management
- Psychology

Other (Specify):

3. How many usability tests have you performed in the last three years?

Select --------->

4. Define the level of your expertise on usability

- ☐ Advanced
- ☐ Above Average
- ☐ Average
- ☐ Below Average
- ☐ Basic
- ☐ No expertise

Part 2/4: About Your Usability Group

Please answer the following questions related to your company. **All fields are mandatory.**

5. What is the size of your company?

Select -------------------->

6. How many years of experience does your company have in usability tests?

Select --------->

7. How many usability specialists are currently working in your company?

Select --------->

Part 3/4: Your Usability Testing Methods & Practices

The following questions are related to the usability testing methods and practices. Try to answer these questions according to usability tests you have conducted. **All fields are mandatory.**

8. Indicate how much the following activities of the usability testing process are important :

None, Low, Medium, High and Very High define the degree of the importance of the activity.

	None	Low	Medium	High	Very high
Define user profiles	☐	☐	☐	☐	☐
Define the usability test team and tester profiles	☐	☐	☐	☐	☐
Conduct a pre-test yourself	☐	☐	☐	☐	☐
Conduct a pilot test	☐	☐	☐	☐	☐
Collect informal comments from the participant after the tests	☐	☐	☐	☐	☐

Review the pre-test & post-test questionnaires with the participant	☐	☐	☐	☐	☐
Add comments of usability testers and observers	☐	☐	☐	☐	☐
Identify patterns and findings from the test results	☐	☐	☐	☐	☐
Prepare a preliminary report	☐	☐	☐	☐	☐
Select video demonstration clips	☐	☐	☐	☐	☐
Produce or update design guidelines	☐	☐	☐	☐	☐

9. How useful do you find the following methods during your usability tests?

(Hold the mouse of the method name to see a short description)

	None	Low	Medium	High	Very High
Contextual inquiry	☐	☐	☐	☐	☐
Critical Incident Technique	☐	☐	☐	☐	☐
Ethnographic study	☐	☐	☐	☐	☐
Interviews	☐	☐	☐	☐	☐
Prototyping	☐	☐	☐	☐	☐
Questionnaires	☐	☐	☐	☐	☐
Scenarios of use (Use cases)	☐	☐	☐	☐	☐
Storyboard	☐	☐	☐	☐	☐
Thinking aloud	☐	☐	☐	☐	☐
Walkthrough	☐	☐	☐	☐	☐
Wizard of Oz	☐	☐	☐	☐	☐
Focus group	☐	☐	☐	☐	☐
Other (Specify) :	☐	☐	☐	☐	☐
Other (Specify) :	☐	☐	☐	☐	☐
Other (Specify) :	☐	☐	☐	☐	☐

Part 4/4: Your Infrastructure & Tools

The following questions are related to the infrastructure and tools used in your company. Try to answer these questions according to **representative** usability tests you conducted. **All fields are mandatory.**

10. How often do you use the following infrastructures to conduct your usability tests? (Please specify other infrastructures that you used to use when applicable.)

	Never	Occasionally	Often

Fixed usability lab	☐	☐	☐
Mobile usability lab	☐	☐	☐
Remote evaluation via Internet	☐	☐	☐
An external lab	☐	☐	☐
No lab / on site	☐	☐	☐
Other (Specify) :	☐	☐	☐

11. How useful do you find the following test materials when conducting usability tests? (Hold the mouse of the material name to see a short description. Only the first three definitions are provided to avoid confusion.)

	Low	Medium	High
Screening questionnaire	☐	☐	☐
Background questionnaire	☐	☐	☐
Pre-test questionnaire	☐	☐	☐
Post-test questionnaire	☐	☐	☐
Test schedule	☐	☐	☐
Non disclosure agreement	☐	☐	☐
Tape consent form	☐	☐	☐
Check list of test environment	☐	☐	☐
Data collection forms	☐	☐	☐
Preliminary report	☐	☐	☐
Final report	☐	☐	☐
Other (Specify) :	☐	☐	☐

12. Please specify the usability tools/software you used during your usability tests
(Check all that apply and indicate the tools you used and that are not listed)

- ☐ Camtasia Studio
- ☐ Clementine Data Mining Tool
- ☐ ErgoLight Usability Software
- ☐ IBM Ezsort & Ezcalc
- ☐ Noldus Observer
- ☐ Ovologger
- ☐ Sigma Stat, Sigma Plot SPSS Products
- ☐ Survey System
- ☐ None

Other (Specify):

Other (Specify):

Other (Specify):

Your Comments

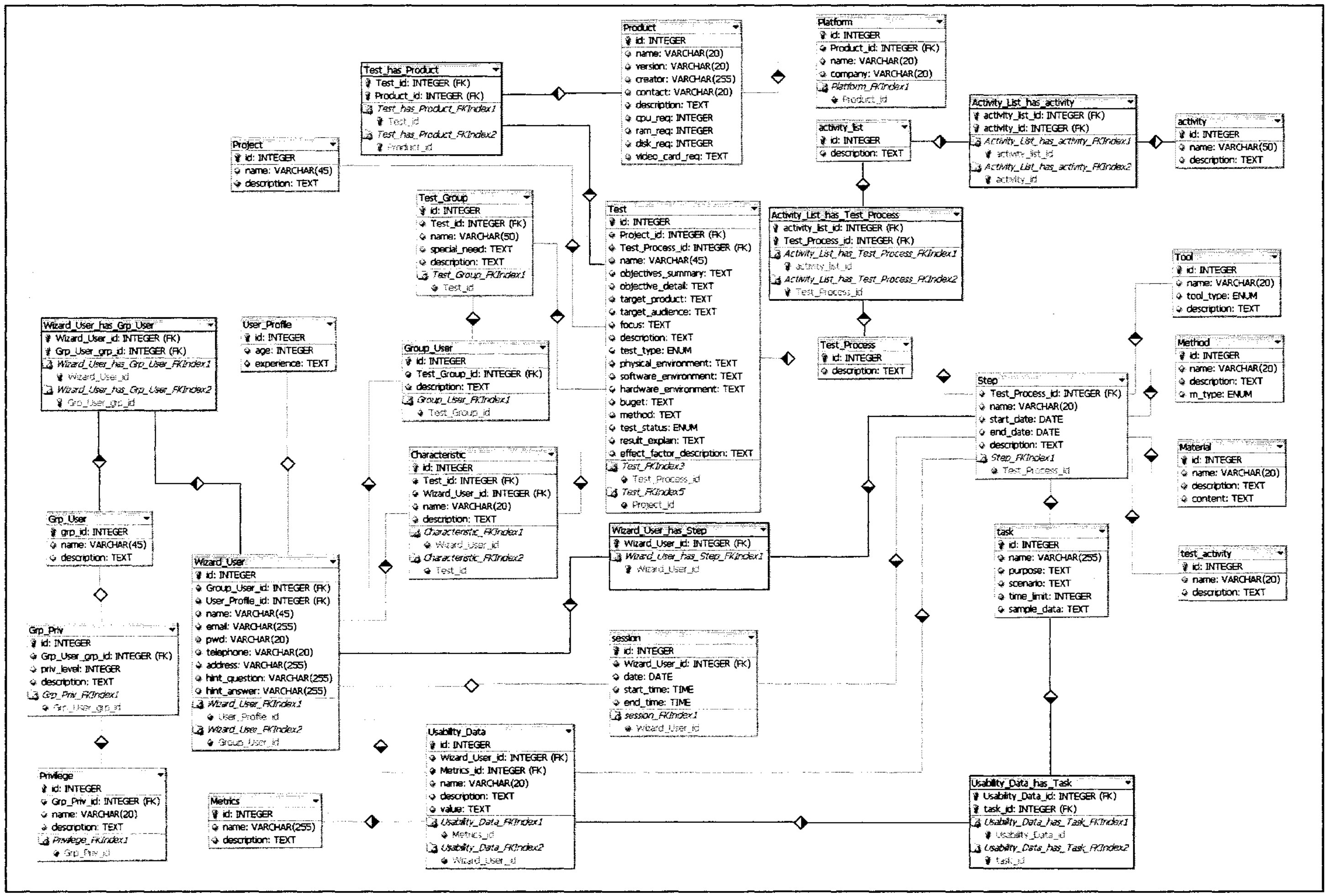
In your opinion, what is required to improve the usability tests in terms of activities, methods and tools? Please feel free to add any comments.

This is the end of questionnaire. **We really appreciate your cooperation.**

Thanks for your time! Submit

Please click on the SUBMIT button to present your data.

Appendix D. WizUse ER Diagram



Appendix E. Samples of Testing Materials

Scenarios

Open the model **FTD_LCO** located on the desktop in the folder UsabilityTest.

Scenario 1: Create a class

1. In the tree diagram on the left, select Logical View | Component Model | «component» Library_User Component | Logical Class Model.
2. Create the class **User_Student** as a specialization of the class **User_Agent** (Relation of heritance).
3. Add the following private attributes in the class **User_Student**:
    - First_Name
    - Family_Name
    - Address
    - Birth_Date
    - ID_Number

    (Don't specify the types of these attributes!)

    Add the following public operations in the class **User_Student**:
    - get_First_Name()
    - get_Family_Name()
    - get_Address()

    (Don't specify the return types!)

Scenario 2: Code generation

1. In the tree diagram on the left, select Logical View | Component Model | «component» Library_User Component | Logical Class Model.
2. Select the Class Diagram **Description_Form.**
3. On the **Tools** menu, point to **Java**, and then click **Generate Java.** What's wrong?
4. Solution: on the **Tools** menu, point to **Java**, and then click **Project specification** and create a new entry in the list of directories, then add the directory C:\j2sdk1.4.0_03\bin\.
5. Check the Rose Log window to view the results of the Java generation, including any errors that occurred.
6. View and edit the generated code
7. Give your opinion about the result of the generation in the given paper (process of generation, quality of code, etc.)

Scenario 3: Draw a sequence diagram

1. In the tree diagram on the left, select Logical View | Behavior Model.
2. Create the sequence diagram 'Login' for use case 'Login' related to the administrator authentication for this sequence.
3. The steps to follow are:
   i. Administrator enters password, the system verifies the password in the database.
   ii. If valid password, the system allows administrative access; otherwise rejects request

(Help: Refer to the activity diagram **Login** see Use case View | Proposed system | Proposed Activity | Administrator Login)

Scenario 4: Documentation generation

1. Generate the 'Summary of packages and class in the logical view' report.
2. Save the report on the desktop in the folder UsabilityTest\Report\ with the name 'Summary_YourName.doc'.

Scenario 5: Identify main elements of the system

1. Identify the components of this system by browsing the component diagram and note them down in the given sheet of paper.

2. In the tree diagram on the left, open Use Case View | Domain Description| Current system | Block Diagram
   Now, Open: Use Case View | Proposed System | Proposed system | Block Diagram

   In the given sheet of paper, note down the main difference(s) in both the block diagrams.

3. In the tree diagram on the left, open Use Case View | Domain Description | Current system | Typical Configuration
   Now, Open Use Case View | Proposed System | Proposed system | Typical Configuration

   In the given sheet of paper, note down the main difference(s) in both the 'Typical configuration' diagrams.

Background Questionnaire

Name: ____________________ E-mail: ________________________

Gender: ☐ Male ☐ Female

Please answer following questions in order to help us get some background information for our further analysis.

1. Education: (Please check out the your highest level of education)
   ☐ Undergraduate student ☐ Graduate student
   If you are graduate student, please list your major area of your undergraduate study.

   ____

2. Computer experience
   How long have you been using computer? (Please check out the option that is fit you, same as the other questions.)
   ☐ 0-3 years ☐ 4-7 years ☐ More than 8 years

3. What is the computer platform you usually use?
   ☐ MS-DOS ☐ Microsoft Windows ☐ Unix ☐ Apple MacOS
   Other:________________________ (Please specify)

4. How long have you been using Rational Rose?
   ☐ No experience ☐ Below 6 months ☐ 6-12 months ☐More than 12 months

5. If you have ever used Rational Rose, what was the main purpose of using this software?
   ☐ Use case design ☐ Data modeling ☐Code Generation
   Other:________________________ (Please specify)

6. If you have ever used Rational Rose, where did you learn?
   ☐ Learn it yourself ☐ Learn from courses ☐ Learn from your project team member
   Other:________________________(Please specify)

7. If you have ever used Rational Rose, what's the version of Rational Rose?
   Please specify all versions you have ever used:______________________________

   ____

8. What is your general attitude of Rational Rose?
   ☐ Dislike ☐ Neither dislike nor like ☐ Like
   If you choose dislike option, please describe the main problem of Rational Rose. (Use as few words as possible.)____

   ____

   If you choose like option, please explain the main reason.

   ____

   ____

9. Please specify the similar software you have ever used besides Rational Rose.

   ____

   ____

Post-test Questionnaire

Name: ______________________

1. Is it easy for you to use this version according to your previous experience?

☐ Yes, I can use it without any problem. ☐ Yes. But some parts of it are new to me. ☐ No. I need to learn how to use this version.

2. Do you think that the icons and menus of Rational Rose are obvious and seemingly intuitive?

☐ Strongly agree ☐ Agree ☐ Neither agree nor disagree ☐ Disagree ☐ Strongly disagree

Explain why: ______________________

____

3. Do you understand the terms and feedbacks used in the software?

☐ Yes

☐ No If no, in which case? ______________________

4. Are the feedbacks clear for you to do your job?

☐ Yes

☐ No If no, in which case? ______________________

5. Do you think the help and documentation are useful?

☐ Strongly agree ☐ Agree ☐ Neither agree nor disagree ☐ Disagree ☐ Strongly disagree

6. Do you think Rational Rose provides the mechanism to prevent you from doing mistakes?

☐ Yes ☐ No

Give some examples to prove your answer ______________________

____

7. Did it easy for you to relate different diagrams, uses case and components?

   ☐ Yes ☐ No

   Give some examples to prove your answer ______________________________

   ____

8. How do you rate the visualization capability of Rose (e.g. the interface layout, navigation facilities)?

Bad		Average		Excellent
0	1	2	3	4

____

Thank you!