The Need for Tools for Usability Testing

Usability quality

As software companies face ever-increasing competitive pressure through the process of development of new applications, they need to define, understand and focus on their core competencies. A main contributor to a company's success is the quality of it's product lines. The term Software quality refer to various perspectives, including reliability and usability.

Software companies prefer to prevent bugs rather than to fix them after they are already introduced to the market. The reason for this is that the expenses of problem fixing after the product is already installed at the customer site are very high, and the costs of loosing the market are immeasurable. This attitude characterizes all software design, and particularly user interface design.

Industry standards are not enough

The common rules for user interface design are based on the means provided by popular development tools and on industrial standards. The tools and the standard facilitate the acquaintance with new software products. However, they do not contribute to the main usability issue, namely, to serve the software features that correspond to a user task, according to the user's expectations. In fact, the most important usability issues are not even in the scope of industry standards.

Not all human factors are well understood

Recently, more and more companies consult experts in ergonomics (human factors) as early as at the specification stage. However, many of the problems presented in a software product are overlooked even by professional user interface designers. Another limitation of consulting services is their costs. Not all software companies can afford to pay for these services.

Problems with traditional usability testing

Although the usability a software product is determined by design, the only means to verify that is indeed usable is by testing.

Traditional usability testing relies on observations of experts in software quality assurance during Alpha testing and on reports of the end users during Beta testing.

The limitations

Typically, the persons who test software in Alpha testing are of technical background. Traditional testing, by persons of technical

of Alpha testing

background, do not reveal many usability problems, especially those that characterize the first use of a new product. Eventually, they attribute the reasons for usability problems to the end user, rather than to the software design.

The limitations of Beta testing

In Beta testing, the end users are cooperative in reporting on those problems which prevent them from completing their tasks. However, end users are not likely to report on a problem if they find a way to work around it. Many usability problems are not identified even in Beta sites, because typically:

Users cannot repeat most failure modes

Users prefer to find ways to work around an operational error rather than to report on it

Users consider operational errors their own fault, rather than design problems.

For these reasons, many usability problems are not identified through the whole product life cycle.

What is required for better usability testing

To overcome these limitations, large software houses raised special department, Usability labs, operated by experts in Human Factors engineering. In these labs, real users are observed by professionals, while they use the real product for completing real tasks. The analysis of usability problems involves various techniques, such as video recording and back tracking and methods such as using questionnaires and "think aloud". Many of the usability bugs are found using these means.

The effectiveness of Usability labs

The operation of the techniques used in usability labs is very expensive and time consuming. Many developers of high functionality Windows applications, including small and medium software houses, are forced to compromise usability issues, because they cannot afford to pay for the services of usability labs. As time and budget resources are always limited, usability labs need to focus on the main problems and to ignore others. Eventually, many software products that were examined in usability labs, including main line products of the leaders in the software industry, suffer from severe usability problems.

Another limitation of the Usability labs is that they do not quantify the usability problems. For example, they cannot provide statistics on the total time wasted because of the user's unintentional press of the Caps Lock or the Alt key.

When to Use

Use ***ErgoLight™*** whenever your requirements include demands for:

Quality	Your customers think that it is important that the Windows application will be easy to use	***ErgoLight™*** identifies design features that are not easy to use and reports on them to the application designers
Costs	It is too expensive to fix the application after it is already shipped to your customers	***ErgoLight™*** collects data on usability problem early at the specification, prototyping and Beta testing phases
Caution	You cannot always identify why a user fails to use a feature	***ErgoLight™*** identifies user failure modes that rarely occur and allows analysis of the failure modes by backtracking
Decision	You do not have an objective measure that supports the preference of a usability solution over the other.	***ErgoLight™*** provides objective measures of the usability problems, in term of the total time waste
Debate	There is more than a single way to provide a user interface for each of the application features	***ErgoLight™*** estimates the benefit of a design feature in terms of time saving and compares this estimate to the total time waste due to unintentional feature activation

From Specification to Deployment

ErgoLight™ is used through the whole life cycle of the application development:

specification . At the specification phase, use ***ErgoLight™*** to specify the user task breakdown and the modes that affect the way the application responds to the user actions

prototype . At the prototype phase, use ***ErgoLight™*** to link the user tasks and the application modes to the actual GUI components

beta testing . At the beta testing phase, use ***ErgoLight™*** to collect data on usability problems

evaluation . At the evaluation phase, use ***ErgoLight™*** to analyze the data and to obtain reports on usability problems, on your recommendations for design changes and on your instructions for the Help Desk personnel

deployment . At the deployment phase, use ***ErgoLight™*** to provide recovery information for the end user and to provide Help Desk information for support centers.

Technology

The technology hereby described is patent pending

Recording the user actions

ErgoLight™ records the user actions when operating a Windows application, allowing back tracking, as well as history based automated analysis of the user confusion.

Identifying instances of user confusion

ErgoLight™ provides both manual and automated identification of instances of the user confusion. Automated identification of user confusion is based on confusion identifiers, such as the user response delay, activation of a Help feature or invoking a Cancel or Undo feature.

Interpreting the user intention

ErgoLight™ provides both manual and automated interpretation of the user intention. The user intention is interpreted in terms of the user task breakdown, which is entered to the ErgoLight™ database at the design phase.

Identifying usability problems

ErgoLight™ compares the user recorded actions to the user's intention, analyzes the matching between the user's intention and the user actions and identifies usability problems of three types:

Problems typical to users new to the application	Tasks that the user could not accomplish using the software application	For example, the user of a word processor wants to move text but fails to either do the Drag and Drop method, or understand the clipboard concept and how to use it.
Problems typical to experienced users	Sensitivity of the user interface to psychomotoric user errors	For example, the unexpected appearance of a dialog box on screen, because of the mouse slip or because of using the wrong shortcut key combination.
Problems that confuse occasional users	Confusion due to global attributes, such as setup parameters	For example, when the user fails to print because s/he is not aware of the “print to file” check box.

On line recovery

ErgoLight™ provides on-line assistance to the end user, based on the results of the problem analysis. This information provided allows the user to resolve certain usability problems,

such as mode errors, and to learn how to avoid error prone operations, such as using the wrong key combinations.

Identifying deficiencies in the user information

ErgoLight™ classifies the records of user confusion by available sources of user information. For example, a designer can extract a report containing all user reports associated with the user's guide.

Evaluating the usability problems

ErgoLight™ provides various means for evaluating usability problems:

Backtracking the user actions, most useful in understanding error modes

Top down review of problems in user orientation, with reference to the relevant sources of user information

Top down review of conflicting Windows controls, with reference to optional design changes

Top down review of mode errors, with reference to optional design changes.

Quantifying the error costs

ErgoLight™ provides statistics of all identified usability problems, thus manifesting the effect of recurrent seemingly minor problems. The user errors are quantified in terms of the user time waste, allowing the application developers to:

Prioritize the usability problems by the error costs

Compare alternative implementations of a user task.

The usability problems are sorted by either the frequency of occurrence or the total user time waste.

Close-loop adjustment to the user terminology

ErgoLight™ provides means for easy adjustment of terms in the task breakdown according to those used by the end users.

Providing feedback to the application designers

ErgoLight™ provides means for adding comments to each usability problem identify, useful for transferring the knowledge gained by the application evaluator to the design team. ErgoLight™ generates default comments, using a knowledge base of user error modes.

Providing Help Desk information

The information obtained by ***ErgoLight™*** is used mainly to identify design problems so that they can be fixed before delivery of the application. Unfortunately, the process of problem fixing is time consuming. Typically, many of the fixes are postponed to the next product release and the product is delivered even before the problems are fixed.

ErgoLight™ provides means to work around the problem for the meantime, until a new version of the application is available:

It provides additional information for the end user, added on the original application

It provides help desk information for customer support centers.

For the Help Desk, ***ErgoLight™*** provides means for adding a comment to each usability problem identify, similar to the comments which the application evaluator adds to reports aimed for design changes. These comments are useful for transferring the knowledge gained by the application evaluator to technical support Help Desk centered. At the Help Desk, ***ErgoLight™*** conducts interactions synchronized with the end user, that allows the technical assistants to easily locate the comment relevant to a particular call for help.

Platforms

The first implementation of ***ErgoLight™*** is on Windows 95.

Stage 1. Definition of the User Interface

Stage 1 is typically conducted at the designer site, by user interface designers, such as system analysts or, preferably, by human factors engineers. At this stage, ***ErgoLight™*** is used as a typical CASE (Computer Aided Software Engineering) tool. The definition stage is typically conducted in 2 steps:

1. User task specification
2. User operation and problem indicator definition

Step 1: User Task and System Mode Specification

Step 1 applies to the specification phase of the product development.

Following common practice of usability engineering

A common practice in User Interface specification is to write down the user tasks. This practice is well known as Task Analysis. Typically, the user tasks are specified hierarchically, using general concepts at the top level, using specific user goals and methods at the interim levels and using detailed description of the operational procedures at the bottom levels. Typically, the user task breakdown thus obtained is used also for defining the Help/Topics feature, used for on-line user assistance.

When using ***ErgoLight™***, The User Interface designer is required to store the task breakdown, which is typically written down as required when using the method of task analysis, in a special database delivered with ***ErgoLight™.***

User task database

ErgoLight™ stores a breakdown of all user tasks supported by the software application in a database, thereby providing a common source of user interface specification.

The process of user interface specification using the task breakdown database is very effective in detecting usability problems as early as at the specification stage.

The data structure is hierarchical, following methodologies of task analysis commonly accepted by the community of human factors engineers. The task hierarchy includes: main tasks, sub tasks, goals, methods, operation and context.

The hierarchical data structure is perceived as intuitive and easy to follow by end users.

Reusing the user information

ErgoLight™ provides a means for converting data from the Windows clipboard to lists in the task break down. This feature is useful to reuse task related information stored in any source of user documentation, such as the user documentation

Reusing the user information saves the overhead of data entry of the task break down, shortening the task break down procedure to few hours.

Step 2: User Operation and Problem Indicator Definition

Step 2 applies to the prototyping phase of the product development.

Prototyping

Prototyping involves the following activities:

Definition of the user operation:
Procedure implementation: association of the steps of the operation procedures to the Windows controls of the product implementation
Definition of the system modes that restrict the applicability of the user tasks.

Definition of the indicators for user difficulty.

Procedure Implementation Definition

In order for ***ErgoLight™ to*** be able to interpret the user actions, the User Interface designer is required to specify the Windows controls associated with each operation step.

Connection by point and click

ErgoLight™ connects each step of an operational procedure to a set of alternative user interface components, such as menu items, buttons and shortcut keys, used to actuate the procedure step. ***ErgoLight™*** provides easy connection, by running the Windows application simultaneously, so that the user interface designer need just to point and click on the component chosen to implement a procedure step.

Connection by point and click is very fast and very reliable, shortening the connection procedure to few hours.

Task verification

ErgoLight™ identifies defects in the user task break down database, including:

Tasks that are not properly defined, namely, those entities in the task breakdown that have only few children, or not at all

User interface components that are "orphans", namely, those components that do not have any corresponding procedure step or parameter in the task breakdown. Typically, the effect of each user interface component should be reflected in the task breakdown. Therefore, the existence of an orphan may suggest that either the task breakdown is not fully specified or that the user interface component is redundant.

Task verification allows detection of defects in the User Interface as early as at the prototype stage.

Mode Specification

In a typical user interface, certain user task are not always applicable. For example, editing a document using a typical word processor is applicable to "open" documents only. A parameter, whose value determines the applicability of a user task, is often referred to as a system mode. Thus, in the example, the applicability of document editing depends on the "openness" system mode.

Windowing the system modes

ErgoLight™ identifies mode errors, if the system mode is "visible" to Windows. Visibility to Windows means that a Windows control, such as a check box or a radio button, is associated with the system mode and that Windows has the handle for this control. For example, ***ErgoLight™*** identifies the "print to file" mode by checking the "print to file" check box of the "Print" dialog box.

At the mode definition phase, the User Interface designer is required to store the dependence of the user tasks on the Windows controls that hold the system mode. The assignment is made by "point and click".

Automated identification of mode errors is useful for on-line recovery from confusing situations, when the application behaves not according to the user's expectations.

Definition of the User Problem Indicators

ErgoLight™ identifies instances of the user difficulties while using the Windows application by Problem Indicators.

Types of Problem Indicators

ErgoLight™ uses two types of problem indicators:

The user time response delay. It is the Test Manager who assigns threshold values for the user time delay, for deciding when a delay in the user response will be considered as a candidate for an instance of user difficulty. The user interface designer is required to assign a default value for this threshold

The user activation of particular Windows controls, used by the user interface designer to implement Cancel, Undo and Help features. For example, the user activation of the Edit/Undo menu item is typically an indicator of a candidate for an instance of user difficulty.

Problem indicators allow computer initiated reporting, essential for capturing many instances of user difficulty that users tend to ignore.

Stage 2: Data Collection

Stage 2 is typically conducted at the user site, such as at Beta sites. At this stage, ***ErgoLight™*** is used as a test controller. The data collection stage is typically conducted in 2 steps:

1. Specification of a test plan

2. Monitoring the user's operation.

Step 1: Specification of a test plan

Step 1 applies to the test setup. Typically, it is conducted by the test manager.

Following common practice of usability testing

A common practice in Beta testing is to group the users according to their role in using the Windows application. ***ErgoLight™*** supports this practice by providing means for assigning setup parameters to user groups. The setup parameters supported are the user time delay threshold and the setup of the level of intervention in case that a putative instance of user difficulty is identified.

Classification of the test results by user groups is essential for analyzing the benefits of design features.

Step 2: Monitoring the user's operation

Step 2 applies to the collection of usability problems at run time.

User initiated problem reporting

While the end user operates the Windows application, ***ErgoLight™*** runs in the background. The end user can invoke a reporting session by activating a special key combination, assigned at design time by the dialog designers.

The user can initiate a reporting session any time during the operation of the Windows application.

Reporting the user intention

Problem reporting is computer instructed. ***ErgoLight™*** prompts the end user to specify the user intention by a sequence of list selections. Each reporting session starts with a list tasks associated with the most recent user action. The end user can select a task from the list, or else, to obtain another list of main tasks, followed by sub task, goal, method, procedure step and context. The sequence of list selections follows the hierarchy of the task break down.

The top-down sequence of list selections make the reporting on a usability problem short and easy to follow.

Understanding the user expectations

ErgoLight™ prompts the user to report on intentions not found in the task breakdown database. The information reported thus is useful:

To learn about useful features not implemented in the software application

To identify defects in the task breakdown stored in the user task database

To exhibit problems in the user information, such as terminology problems, that prevent the users from understanding the actual capability of the software application.

Exceptional user expectations are typical to users new to the Windows application. The information acquired is similar to that obtained using the traditional "think aloud" technique, but, due to automation, the reporting is much more efficient.

Reporting problems in the user information

ErgoLight™ prompts the user to report on operational procedure they could not find in the user information. The user is asked to specify the source of information s/he tried, such as the User's Guide, the on-line Help and the Tutorial.

The classification of user information problems facilitates the decision making regarding fixes and the distribution of the problem reports to the persons that should fix the problems.

Tracing the user actions

ErgoLight™ runs in the background of the Windows application, captures the user actions and stores them in the User Action database.

The user actions are used for automated and manual analysis of the reasons for the user confusion, at run time and at the evaluation phase

Identifying instances of user confusion

Using the problem indicators specified at the design phase***, ErgoLight™*** analyzed the user actions on line, identifying situations of the user experience of difficulty in operating the Windows application. ***ErgoLight™*** initiates a dialog with the end user whenever an instance of possible user confusion is indicated.

The automatic detection of user confusion allows to reduce to minimum the interference with normal operation flow.

Dialog control

Each time a possible user confusion is indicated, ***ErgoLight™*** prompts the user to either start a reporting session or to resume normal operation. The user always has full control over the operation flow, reducing the interference with normal operation flow down to the barest minimum.

The user control over the operation flow is essential for goal driven operation.

Reducing false alarms

False alarms are situations when a user difficulty is wrongly identified, namely, when a problem is indicated although the user does not feel that s/he has any problem at all. Example of false alarms are:

The activation of a Help feature for exploring the behavior of the Windows application at the initial learning phase

The activation of a Cancel control in a process of learning by "trial and error"

Response delay that occur when the end user pauses operation, because s/he was doing something else.

False alarms are typical of the phase of initial learning the Windows application and they interfere with the user learning process. ***ErgoLight™*** allows the user to prevent false alarms by disabling the problem indicators. ***ErgoLight™*** provides three levels of false alarm prevention.

A testing administrator can disable the problem indicators for a whole session

The end user can disable the problem indicators for a sub session

In normal operation, when the problem indicators are enabled, the user can bypass them using special bypass key combinations, assigned at design time by the dialog designers.

The hierarchy of three level false alarm prevention allows fine tuning the computer intervention, to balance between the need for maximal reporting opportunity and minimal interference with the user learning process.

On-line user assistance

ErgoLight™ provides to the end user various services of on-line assistance, including:

Automated back tracking of recent user actions, that the user can compare to his intended action

Automated analysis of the reasons for unexpected behavior of the Windows application, including a list of attributes that could affect the behavior of the software application.

The on-line support service reinforces the users to report on problems they encounter.

Stage 3: Evaluation

Backtrack

ErgoLight™ identifies situations of user confusion that rarely occur. The identification of such situations is provided by on-line detection and backtrack of the user actions. At the analysis stage, the evaluator can identify those instances which resulted in user confusion that significantly hampered the user performance and to backtrack the user actions. ***ErgoLight™*** identifies possible reasons for the problem, if the reason is sensitivity of the software application to psychomotoric user errors, or to mode errors.

Many of the severe usability problems are hard to detect, because they are rare and because they are hard to retrieve.

Problem classification

ErgoLight™ classifies the reported problems by their type, as follows:

Exceptional user intention: The user intention is not found in the task break down. Exceptional user intention problems include user tasks not supported by the application, user tasks omitted from the task break down or user tasks specified using terms different from those which the end user uses

Unexpected application response: The response of the Windows application to the user action was not as the user has expected. Unexpected application response problems include problems due to the sensitivity of the user interface to the user errors and problems due to mode errors. Unexpected application response is typical to users that have some experience in using the application

Problem classification is used to derive different reports, each aimed to those persons that should take care of fixing the problems.

Statistics

ErgoLight™ collects statistics regarding all identified usability problems, and quantifies the usability problems in terms of the percentage of total time waste.

The total time waste is a measure of costs. Using statistics, a designer can:

Prioritize the usability problems according to the costs saved by fixing them

Compare alternative implementations of a design feature

Controversial design decisions can be judged using objective measures, based on their real costs.

Adjusting the application terminology

ErgoLight™ provides integrated "Search and Replace", specialized for controlled changing of terms used in the user task breakdown by those that the end users used in order to describe their intention when they encounter a usability problem.

Much of the confusion that end users typically encounter is due to the use of technical terms instead of user terms.

Knowledge transfer

ErgoLight™ provides means for the test evaluator to transfer the knowledge which s/he gains while going through the usability problems to the designers and to the Help Desk personnel. The knowledge is recorded as comments to the usability problems that can be organized later on by the type of usability problem and the user task.

Benefits of Automation

Comprehensive detection

Problems of sensitivity of the user interface to user errors are hard to detect, because users typically prefer to work around their errors rather than to report on them. ***ErgoLight™*** automatically detects and reports on a wide range of problems that are typically ignored in traditional software testing.

ErgoLight™ identifies situations of user confusion and initiates a reporting session even before the user is aware of the fact that a usability problem has occurred, or that it should be reported to the application designers.

Problems in procedure knowledge

ErgoLight™ identifies situations in which the user is not sure about the procedure that should be used to accomplish a task. ***ErgoLight™*** identifies such situation by problem indicators, such as the user response delay or the activation of a Help feature.

Sensitivity to unintentional actions

ErgoLight™ identifies problems that users are likely to ignore, including those relating to the sensitivity of the user interface to user errors. Consider, for example, what happens in case of a key slip. The result may be the activation of an unexpected dialog box, unintentional mode change or, even worse, activation of an undesired feature, such as erasing data. ***ErgoLight™*** identifies these instances of user difficulty and reports on them as sensitivity problems.

Backtrack

ErgoLight™ identifies situations of user confusion that rarely occur. The identification of such situations is provided by on-line detection and backtracking of the user actions.

Avoiding User's Resistance

User control

ErgoLight™ was designed carefully to provide user control in all operational situations. The main concern is that users might avoid usability testing if it interrupts the fluent operation of the software application. Another concern is the integration with known learning schemes, including learning by reading from the Help screens and learning by "trial and error". At the learning stage, the problem indicators may be invoked frequently. Users might avoid usability testing at the learning stages to avoid frequent interruption of the fluent learning processes.

Filtering

ErgoLight™ allows test managers to disable any of the problem indicators

Bypassing

ErgoLight™ provides bypass controls, allowing the user any time during the operation to bypass any of the problem indicators

Fluent operation

Whenever a problem is indicated, ***ErgoLight™*** allows the user to resume normal operation rather than to report on the problem

User initiation

ErgoLight™ supports user initiated reporting on usability problems they encounter, which may be used also if a relevant problem indicator is disabled.

On line recovery

ErgoLight™ provides on-line assistance to the end user, based on the results of the problem analysis. This information provided allows the user to resolve certain usability problems, such as mode errors, and to learn how to avoid error prone operations, such as using the wrong key combinations..