Abstract: The present paper describes the relation between software quality and interface usability. A research was made on the GUI of Monita Time - software for working time control and management. The experiment has been carried out using a video screen capture. The participants in the experiment, students in the Technical University of Sofia, specialty Engineering Design, performed typical test tasks. The results confirm the opinion that the interface usability evaluation should be a part of the efficient software quality assurance process.

Keywords: usability evaluation, human-computer interaction, ergonomic GUI, software quality.

1. INTRODUCTION

The graphical user interface (GUI) is as important factor in the success of a software project as its embedded functionalities. A software product could be useful, but it will not be used, if it is not sufficiently usable. Usability becomes increasingly marketable and recent observations in the field show that the benefits of increased usability exceed the costs. So, it should not be surprising that lately the code the software developers write for the user interface of a given software product could be equal to the half of the whole code for a project, taking a comparable amount of development effort, which is a prove of the importance of popular program GUIs.

2. SOFTWARE QUALITY AND INTERFACE USABILITY CRITERIA

The presented experiment is a consequence of an attempt to broaden the software quality assessment by including additional usability criteria for the GUI evaluation.

2.1. Usability in the SW Quality standard ISO/IEC 9126

ISO/IEC 9126 which categorises quality from a user perspective is the relevant international standard which defines the software quality characteristics and sub-characteristics (Fig. 1), as follows:

- Functionality - Accurateness, Suitability, Interoperability, Compliance, Security,
- Reliability - Maturity, Fault tolerance, Recoverability,
- Usability - Understandability, Learnability, Operability,
- Efficiency - Time behaviour, Resource behavior,
- Maintainability - Analyzability, Changeability, Stability, Testability,
- Portability - Adaptability, Installability, Conformance, Replaceability.

All these aspects try to summarize the criteria for software quality evaluation. However this standard treats usability only in terms of: 1. understandability, that bears on the users’ effort of recognizing the logical concept and its applicability; 2. learnability, connected with the users’ effort for learning its application; 3. operability, in the sense of users’ effort for operation and operation control.
Unfortunately, yet many software development companies in Bulgaria do not pay enough attention to the more specific meaning of usability, which is discussed in ISO 9241-11.

2.2. Usability in ISO 9241-11

This standard explains how to identify the information that is necessary to take into account when specifying or evaluating usability in terms of measures of user performance and satisfaction. In this standard the three components of usability are defined as:

- **Effectiveness** - the accuracy and completeness with which users achieve their goals,
- **Efficiency** - the resources expended in relation to the accuracy and completeness with which users achieve goals,
- **Satisfaction** - the comfort and acceptability of use.

The first two aspects of usability are partially evaluated according the software quality criteria in ISO/IEC 9126, but especially the third component - user satisfaction, which is very important for the GUI quality from the user's point of view [2], is not included in the quality evaluation process.

3. EXPERIMENT

The presented experiment shows the benefits that the integration of diverse usability aspects in the development and assessment process can bring to the software quality.

3.1. Experiment Description

The tested object is Monita Time – a Bulgarian interactive system with a large database for working time control and management using access cards in small and middle-size companies, which gives reports about the working time, overtime work and etc. The test participants are 8 students from the course in Usability Engineering, Engineering Design specialty at the Technical University of Sofia - 4 novice and 4 trained users.

In general the sequence of typical tasks that should be performed when using Monita Time is to launch the software, search different information about certain em-
ployees and make changes in the data base, prepare and/or print reports and quit the application. So, these were the tasks performed by the test participants and were video captured simultaneously.

The usability measurements accomplished during the usability test are based on the ISO 9241-11: effectiveness, efficiency and satisfaction measures (Table 1).

User performance for key system features is admitted to be conferred to the following statement: “trained operators should be able to create typical data reports with at least 95% accuracy on first attempt in 1 minute”, which is estimation for the task performance of expert users. In addition, user satisfaction should correspond to a minimal level: “a mean score on the rating scale for satisfaction, greater than 3 of 5” or “at least 90% of the users are satisfied with the system feature”.

More information on quality in use requirements was obtained on the basis of specific measurements and comparison to the usability specification of the key system features. The specification includes not only the usability components – effectiveness, efficiency and satisfaction, but also the intended context of use and actual context of evaluation: usage of help function, executing the task for the first time etc.

3.2. Experimental results

The table below should be regarded as a matrix for complex usability evaluation. It contains the values obtained as results from the video captured test.

<table>
<thead>
<tr>
<th>Usability objective</th>
<th>Effectiveness measures</th>
<th>Efficiency measures</th>
<th>Satisfaction measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meet needs of novice users</td>
<td>Percentage of tasks completed successfully on first attempt 23 %</td>
<td>Time taken on first attempt 42 sec.</td>
<td>Rate of voluntary reuse 51 %</td>
</tr>
<tr>
<td>Meet needs of trained users</td>
<td>Percentage of tasks completed successfully 67 %</td>
<td>Relative efficiency compared with an expert user 65 %</td>
<td>Rating scale¹ for satisfaction with system features 4</td>
</tr>
<tr>
<td>Learnability</td>
<td>Number of functions learned 3 of 5</td>
<td>Time to learn 20 min</td>
<td>Rating scale² for ease of learning 3</td>
</tr>
<tr>
<td>Minimization of support requirements</td>
<td>Number of accesses to HELP 6</td>
<td>Productive time 10 min</td>
<td>Rating scale¹ for satisfaction with support facilities 2</td>
</tr>
<tr>
<td>Error tolerance</td>
<td>Number of user errors tolerated 26 %</td>
<td>Time spent on correcting errors 5 min</td>
<td>Rating scale³ for error handling 2</td>
</tr>
</tbody>
</table>

¹ Rating scale: from 1 – “not satisfied at all” to 5 – “very satisfied”
² Rating scale: from 1 – “very difficult” to 5 – “very easy”
³ Rating scale: from 1 – “very bad” to 5 – “excellent”
As it can be seen in the table above, most of the values are not satisfying in the aspect of high quality software. The main problem consists in the confusing way of presenting the information content and in terms of GUI that means that there is an illogical information hierarchy, non-intuitive graphic elements and etc. This could be summarized within the following: In fact the GUI is not available for the most of the users, although GUI is present.

The simple presence of a feature as an element from the GUI is a necessary but not a sufficient condition for the software to be usable. The key system features should be presented in an appropriate and explicit form. A usable software can be achieved only if the GUI is regarded as a separate and essential part of the software product that is been developed. This is also the way to obtain a high software quality respectively users’ satisfaction.

4. CONCLUSIONS

The experiment results suggest the idea that the software products should no more be regarded as a collection of system features, but rather as a complex object of development and design. Unfortunately GUI development in Bulgaria is still considered as unimportant in a big number of small software companies. The field of study called "Human-Computer Interaction" (HCI) claims that the GUI development should be more often allocated sufficient time in the early phases of the development schedule. In this sense there is a need to educate all programmers about HCI issues and to integrate GUI development and assessment into the software development and quality assurance process.

5. REFERENCES