UDC: 004 Computer Science

On the Usability of Social Software Systems: A Preliminary Study Using Elgg

Lenandlar Singh
Lecturer, Department of Computer Science, University of Guyana, Guyana, South America

Abstract
This paper reports on a Usability Evaluation conducted on a deployment of the Social Software Elgg at the University of Brighton: Community@Brighton available at http://community.brighton.ac.uk.
Community@Brighton is a social software system installed at the University of Brighton to provide a forum for social interactions among students and staff at the University of Brighton. Community@Brighton is an instance of the social software Elgg with the present deployment using Elgg Version 0.7. A usability test was conducted on the use of the features and functionalities of the system.

Five (5) lecturers from the School of Computing, Mathematical and Information Sciences at the University of Brighton participated in this study. Each participant performed a number of browsing based and hands-on tasks using the system. The design of the study was centered on the use of the usability testing methodology together with the think aloud technique for usability evaluation.

There were four (4) dependent variables: task completion rates, number of defects, number of assists and problem counts or number of errors per task. A subjective questionnaire was also administered at the end to obtain qualitative exploratory data about users' satisfaction with the system.

A total of 90 defects were identified from the evaluation of which 13 were identified on the Homepage of the system, 39 from browsing based activities and 38 from hands-on tasks. The problems were placed into one of four (4) categories namely navigation – 19 defects, Labeling -20 defects, general heuristic – 21 defects and functionality – 30 defects.

Keywords: *Social Software, Usability, Usability Evaluation of Social Software, Elgg*

1.0 Introduction

The motivation and inspiration for undertaking this project was as a result of a number of different but related ideas and issue, some of which are personal while others are more ranging, with wider potential impact and implications.
One of the main reasons for undertaking this project arises from the author's interest in investigating the use of a free and open source software system to support teaching and learning. Specifically, the author wanted to explore the use of social software application for this purpose. The social software Elgg was the application of choice for this project with a specific instance of it, Community@Brighton, explored.
Community@Brighton was a convenient instance of the software to work with since it is deployed at the University of Brighton where the study was based. In addition, support for this project was guaranteed by the group that deployed and manages Community@Brighton.
From anecdotal evidence, Community@Brighton was noted for being problematic for it's user in a number of ways. Specific issues identified by users of the system seem to be related to the general

understanding of the system and general encountered with the user interface of the system. These observations motivated the study further and lead to the definition of the study itself.
Further, it was thought that the results of a study of this nature would be useful to guide and inform the continued design and development of the parent project itself, Elgg, through this initiative from the Community@Brighton. Specifically, feedback about issues discovered with the user interface of the system would provide for more informed design decisions about the interface.
It is the believed that this study will provide important feedback about the suitability and readiness of this software as a course support tool for teaching and learning.

1.1 Product Description

Community@Brighton – http://community.brighton.ac.uk is described as a 'social networking system for students and staff at the University of Brighton' [1]. Koskinen (2006) [2] describes a social networking system as 'an umbrella containing a mixture of technologies, internet services, and software'. Community@Brighton is powered by the Social Software system Elgg [3] and is currently deployed using version 0.7 of Elgg. This version of the system was tested.
Elgg is described by it's developers as 'an open source social platform based around choices, flexibility and openness: a system that firmly places individuals at the centre of their activities' [3]. Elgg was originally started by Ben WerdMuller and Dave Tosh [3].

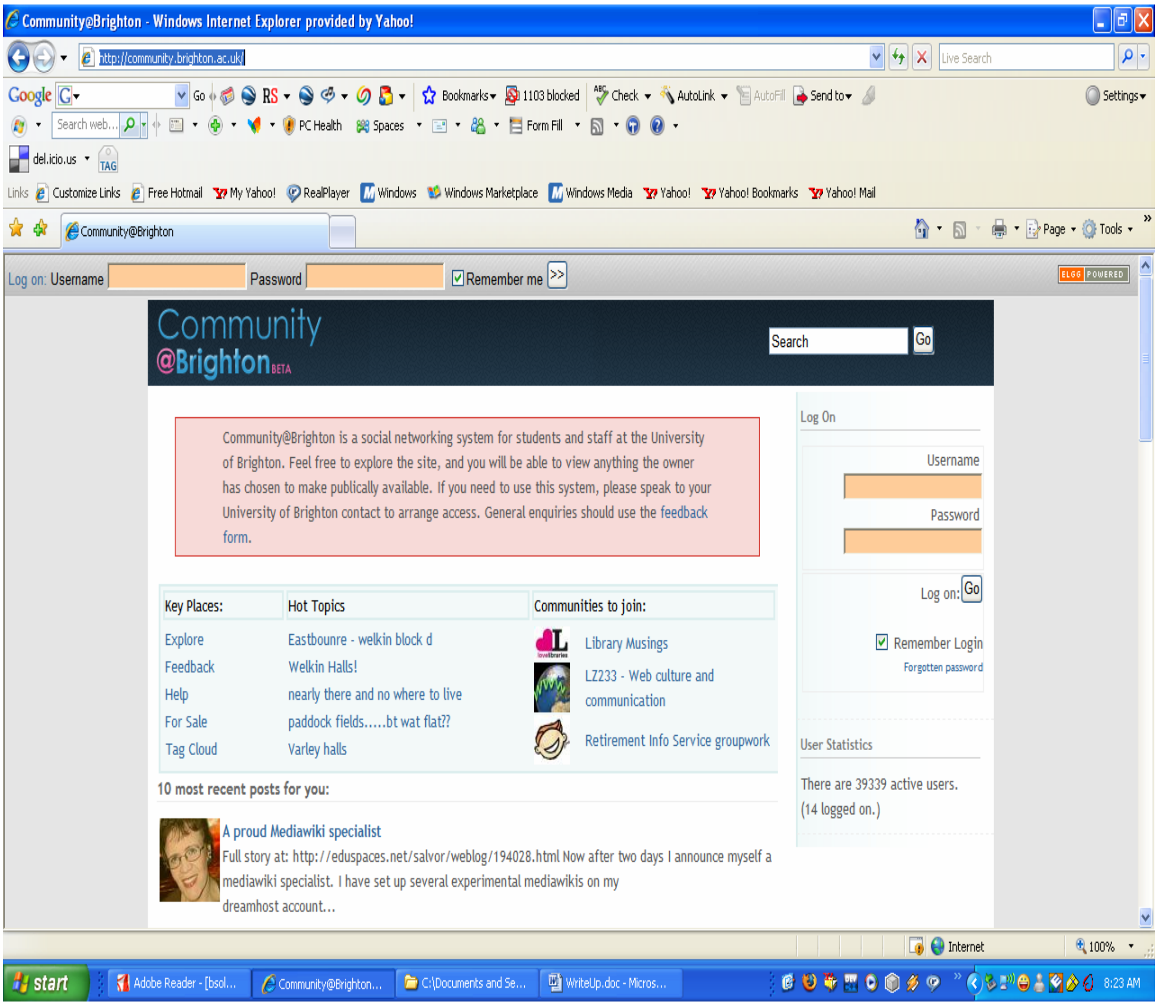


Figure [1] Shows the Homepage of Community@Brighton

The Elgg social software platform is designed using the LAMP (Linux, Apache, MySql, and PHP) Platform. However, an Elgg-based system can be deployed under Microsoft Windows Operating System as opposed to Linux. Elgg is licensed under GNU Public License Version 2 [4].

Elgg, in general, and Community@Brighton, specifically provides the following main facilities for its users:

- Blogs
- Communities – to create new and join existing
- File Uploads
- Networking with Friends
- Creating Presentations
- RSS Feeds
- A Personal Profile

Additionally, the system allows for the integration of a number of plug-ins allowing custom features to be added as needed.
The user population of Community@Brighton is currently very broad but includes mainly, staff of the Learning Technologies Center and students and lecturers at the University of Brighton. Anyone with access to the internet and a web browser can view the system but only registered users from the University of Brighton can add, delete or edit content.
Generally, the entire user interface of the system was evaluated by participants through browsing-based activities. However, specific hands-on tasks related to course support were carried out by the participants as discussed below.

2.0 Literature Review of Social Software Systems Evaluation

According to Preece (1993) [5], the choice of a particular evaluation method depends upon various factors including exact purpose of evaluation; stage of design and development at which it is carried out; the question of ecological validity (permissibility of biases in data collection); the external limitations imposed on the evaluation process (time constraints, development cycle, cost and availability of equipment/expertise).
This wide ranging set of guidelines fuels the on going debate about the 'right' approach and methodology to take when conducting a usability evaluating a software system. More so with the appearance of socials software systems, characterized by its sheer magnitude of use, diverse population of users with differing demographics and a different approaching to the use software systems, there is a need to carefully select an appropriate methodology for usability testing. The need to revisit the approaches used to evaluate these systems takes on new levels of importance.
Social Software systems are likened to groupware systems, with diverse sets of characteristics and interactions between its users. It is believed that 'the ever blurring boundary between work and everyday life is broadening the concept of context and multitasking with multiple users and multiple tools becomes prevalent. All these factors render the task of evaluating social software extremely challenging. The high incompatibility between group activities and usability lab environments calls forth field and longer-term evaluation as well as adaptation of existing usability evaluation methods (UEMs) and metrics to cover most aspects of collaborative experience'[6]. Further, it is suggested that effectiveness and efficiency 'may no longer be significant quality attributes for social software supporting unstructured tasks' and therefore 'user satisfaction which can be gauged by many different factors, becomes the main concern' [6].

The methodology used in this study is the think aloud, usability testing methods, conducted under usability test conditions in a laboratory. This was a suitable methodology from the perspective that there was an existing system in place. Available resources and time constraints also influenced the use of this method.
It is important to note that an 'empty and dead system' was evaluated, in the sense that the participants were not interacting with the system over time and in their work environment, creating real interactions or content and as a result, many interactions were static. This contradicts the way social software systems are meant to be used. Social software systems are not designed necessarily

for use by one person in nature and this is exactly what took place during the evaluation of the system.
Whether the approach used in this study was good enough is therefore a point for discussion and should focus on the dynamics and characteristics of social software systems as its core in order to validate the usefulness of the results of the study.

3.0 Methodology

The test was performed to evaluate in general, whether the system under study – Community@Brighton can be used as a course support tool for lecturers in the School of Computing, Mathematical and Information Sciences, University of Brighton. Specifically, the system was tested for usability and usability criteria:

- **usability** - the degree to which a (the) system assisted the user in completing a task, Nielsen (1998) [7]
- **efficiency –** how the product supports users in carrying out their tasks, Sharp et al. (2006)
- **effectiveness –** user performance of the specific tasks in terms of task completion and accuracy
- **satisfaction –** the extent to which the users accept the system and are satisfied with their progress on completing a task by using the system, Sharp et al. (2006) [8]
- **understandability –** how easy it is to understand what the system does, Lauesen (2005) [9]

Investigating user acceptance of the system was an important objective of this study. Indeed, the Technology Acceptance Model (TAM) speaks to the usability and usefulness of a system as a critical factor in acceptance of the system, Chan & Teo (2007). [10]
The test focused on two (2) main areas: browsing-based activities and task-based activities. The Homepage of the system was explored by the users through browsing-based activities. Additionally, participants were asked to browse each of the identifiable components on the homepage by visiting the respective pages where links from the homepage took them.
The second set of functionalities that were tested with was based on participants carrying out specific tasks by directly interacting with a number of components. The following components along with their respective sub-components were explored:

- Blogs
- Resources
- Presentations
- Networks
- Files
- User Profiles

The main reasons for focusing on the task-based components are that they were thought of as being the components that were important for course support. As a result they, are more likely to be used by participants in their daily work, were they to adopt the system for course support.
Browsing-based activities were conducted to investigate general navigational patterns of the system by participants and their understanding of the components they encounter during browsing.

3.1 Method

3.1.1 Participants of Study

The user population for this study was Lecturers from the School of Computing, Mathematical and Information Sciences, University of Brighton. Four (4) Senior Lecturer and One (1) Principal Lecturer participated in this study. Four (4) of the lecturers were male and one (1) female. All participants were required to have:

(a) basic computing experience
(b) used a course support tool before
(c) general understanding of social software but did not use Community@Brighton before
(d) agreed to have their sessions recorded and saved to a storage media

Participants were asked to participate in this study by the evaluator and represent a sample of the user population that was available to the evaluator at the time of the study. This was not a random sample but an 'available' sample since the study was conducted during the summer vacation period and most lecturers were away on holidays or just not present at the University.
However, there is no reason to believe that this is not a close to representative sample of the user population. Additionally, research has shown that approximately 80% of usability problems can be discovered by 5 participants in a study, Nielson (2000) [11]. Prior to this, Virzi (1992) [12] and Nielson & Landauer (1993) [13] provided some evidence to support the idea of using 5 users in usability testing and as one which will reveal approximately 80% and 85% of usability problems respectively. On the other hand and looked at from another perspective, Lindgaard & Chattratichart (2007) [14] provides some tangible evidence suggesting that tasks selected for usability testing may have a higher correlation with the percentage of problems found than the number of users selected for usability tests.

Table [1] Shows characteristics of Participants

Participants	**Gender**	**Age**	**Education**	**Occupation /Role**	**Professional Experience**	**Computer Experience**	**Product Experience**
JW	Male	40-60	MSc	Senior Lecturer	>10 Years	>10 Years	None
ME	Female	40-60	MSc	Senior Lecturer	>10 Years	>10 Years	None
DH	Male	40-60	PhD	Head of Division	>10 Years	>10 Years	None
RH	Male	40-60	MSc	Senior Lecturer	>10 Years	>10 Years	None
RG	Male	40-60	MSc	Principal Lecturer	>10 Years	>10 Years	None

3.1.2 Context of Product in Use

The system was evaluated in the computing laboratory if the School of Computing, Mathematical and Information Sciences, University of Brighton. This is not a specially designed usability laboratory but was converted to one for the course of the test. Also, this is generally not the context in which the system would have been used by the participants of the test.
Participants would normally use this system from computer systems installed in their offices or at home. More importantly, participants would have never used this system as a means for supporting courses they teach. Instead, the features provided would have been used for other purposes. For e.g. Blogs were used but not in the context of course support.

3.1.3 Tasks

Description of task scenario: there were three (3) main categories of tasks that participants had to perform. The first included participants to observe the homepage of the system and describe verbally the elements on it without clicking. Additionally, this task sought to elicit the user's initial impressions and general perception of the homepage and the system. The second task required the participants to browse each of the elements on the homepage and their sub-pages, indicating as they navigate their understanding and expectations of their actions. The third task required the participants to complete seven (7) hands-on task-based activities. These tasks and the scenarios around which they are based are listed in ***Appendix A***.
The participants were simply given a list of all the tasks with guidelines for describing their experience during browsing. For the hands-on based tasks, description included a scenario around

which each task revolves. A brief verbal explanation of each task and sub-tasks were given to the participants.
These three broad categories of tasks were completed by the participants and are as follows:

1. Description of the homepage elements
2. Description of homepage elements and their sub-elements by navigating or clicking through each element
3. Hands-on activities to create and use specific components of the system

Why tasks were selected:

1. Task 1 was designed to test users general understanding of the terminologies used on the homepage and its general structure.
2. Task 2 was a direct extension of task 1 and was designed to test the participants understanding of the architecture of the system, navigation structures, terminologies and functionality.
3. Task 3 related directly to the tasks that users will do with the system. This task was designed to test efficiency and effectiveness in carrying out the tasks.

Task source: Dumas & Redish (1999, p.160) [15] notes that 'usability testing is a sampling process' and that it is impossible to test every possible task users can do with the product'. They further went on to list four (4) conditions from which tasks can be sampled, namely:

1. Tasks that probe potential usability problems
2. Tasks suggested from your concerns and experience
3. Tasks derived from other criteria
4. Tasks that users will do with the produce

The above listed conditions were used as a general guideline in compiling the list of tasks for the test. Specifically, sources for selecting given tasks are listed as follows:

1. Tasks that would potentially cause a breach of general Usability Guidelines were subjectively selected by evaluator
2. Ad hoc usability issues encountered while using the system by evaluator
3. Feedback from Dr. Jon Dron [16] – a frequent user of the system

Task data given to participants: the task data for all tasks and sub-tasks were given to participants on an instruction sheet which can be seen in ***Appendix A***. Also, the evaluator was available to answer any question posed by participants or clarify instructions.

Task performance criteria: a number of criteria were used to measure success and includes:

1. accurate description of an element or component
2. accurate prediction of a behavior or functionality of a component
3. successful completion of a hands-on task or activity by observing the relevant feedback for successful completion of a task

3.1.4 Test Facility

Intended context of use: an internet connected computer with a browser installed. This could be at home, or in an office or workplace setting where a computer can be used and the internet accessed.
Context used for the test: the usability test was conducted in Room W 204 of the Computing laboratory of the School of Computing, Mathematical and Information Sciences, University of Brighton. Two (2) of the existing networked computers were configured to allow the participant and the test administrator to work side by side.
Participants worked alone but with the test administrator at a neighboring computer waiting to answer any queries that may have arisen during the test and at the same time asking relevant questions about why participants were undertaking certain actions in an interactive way.

The test administrator computer, even though just beside the participant's computer, was set up to 'remotely' view the participant's interactions using the Morae Remote Viewer Software. Participants were asked to think aloud while doing the tasks. The Morae [17] usability software suite tracked and recorded the entire session which was saved at the end of the last task.
For 2 of the 5 sessions recorded, the computing system was running noticeably slower than the other sessions. Possibly reason for this could have been that the Morae software needed more resources to run than was available, hence slowing the system down. Another reason could have been that the computer being used for the sessions was not the most suited for the Morae software running in a real environment. This may have affected the flow of interactions by these 2 participants.

Intended Context of Use: Community@Brighton is intended to run on a computer with an internet connection and a web browser installed.

Context of use: The test was run using an internet connected PC with access to Community@Brighton at http://community.brighton.ac.uk. The PC ran Windows XP operating system with Service Pack 2 and fully updated. The PC was a Dell Computer with 2.4 GHz Pentium 4 processor and 512 MB of RAM. The web browser used was Internet Explorer 7.0. The data logging software used was Morae v 1.2.0.

3.1.5 Test Administration Tool

Sessions were recorded and videotaped using the Morae Usability Software. The Morae Software was set up to capture combined pictures of the screen and the participant's interaction with the interface along with view of the participants through a web camera connected to the software. Additionally, output from users thinking aloud i.e. users verbal output was recorded using a microphone that was connected to Morae. Keyboard presses and mouse clicks and movements were also logged using Morae.
Morae's Remote Viewer was also set up on the test administrator's computer and was used to view the users actions 'remotely'. The Remote Viewer also allowed the test administrator to place markers on specific events of interest as they occur without interrupting the participant.
At the end of the session, a subjective questionnaire (see Appendix B) was administered to the users by the test administrator. This questionnaire was specifically design to solicit participant's satisfaction in using the system. Questions from the questionnaire were verbally asked by the test administrator and participant's verbal comments were recorded using the Morae software via the microphone.

3.1.6 Design and Approach of Evaluation

Sharp et al. (2006, p.591) [8] states that there are three (3) main usability approaches, namely:

- usability testing
- field studies
- analytical evaluation

The approach used in this study is the usability testing conducted in a regular computing laboratory converted into a 'usability laboratory-like' setting for the purpose of this study by installing the required equipment and software. The usability test was in the form of a controlled experiment, Sharp et al. (2006) [8]. The test administrator prepared most of the tasks described in Appendix A. However, participants were given some freedom with the browsing-based activities.
During user testing, Sharp et al. (2006) [8], participants were asked to use the think aloud technique to provide verbal feedback for their actions when undertaking tasks. Additionally, the test administrator interacted with the participants during the session, asking questions when it was thought to be important to ask. Participants were encouraged to raise queries they may have had

with the tasks. This approach allowed the test administrator to take advantage of opportunities to further investigate issues that were not otherwise thought of.

The test environment and the overall suggested time required to complete the test were decided by the test administrator. Data from participant sessions were recorded and logged using the Morae Usability Software. All data generated from the Web Camera, Microphone, Keyboard and mouse clicks were logged.

There were 2 main dependent variables: number of defects or problem counts while browsing the interface of the system or completing a task, and level of difficulty or severity of problem on tasks. The severity of each defect or problem was derived using a customized format of defects classification as proposed by Lauesen (2005, p.12) [9]. Severity was classified as follows: minor problems, major problems, task failure subjective questionnaire (Appendix B) was administered at the end of the tasks to obtain qualitative satisfaction data.

3.1.6 .1 Procedure

Upon arrival participants were reminded that they had been asked to participate in an evaluation of an instance of the social software Elgg deployed at the University of Brighton. The instance of this social software is called Community@Brighton.

Participants were told that by taking part in this evaluation, they would help us find some of the problematic issues of this software and as a result their input will inform the re-design of the software to make it easier to use and learn. They were told that this evaluation was not done to test their performance or skills in any way but to test the software and that they should not be worried about making mistakes or carry out tasks that might show up their inexperience with using the software.

The were told that they would carry out typical tasks expected of a user of the system and that their entire session using the system would be recorded using the software Morae which will record their keyboard and mouse clicks, their speech through the microphone and video through the use of the web camera. Participants were told that all information recorded during this session would be used solely for the stated purpose of the evaluation for which the agreed.

The test administrator opened up the web browser and loaded Community@Brighton and cleared the cache and cookies repository to get rid of any previously used logged in information not related to this participant. Participants were not all tested on the same day but tested on a separate day respectively.

The test administrator stayed in the same general area with the participants using a nearby computer to remotely view the participant’s actions as they use the system and interacted with the participants directly. There were no time limits on the tasks. However, if it was felt that participants were spending too much time trying to complete a given task, the test administrator intervened and made a decision to help or to ask the participant to move on to the next task.

Each participant was allowed a maximum of 2 hours to complete all the tasks but were told that they should not feel obligated to complete all tasks and that they can leave before the end if they so desire. After the last task, participants were asked to asked to complete a subjective questionnaire (see Appendix B) which the task administrator managed by reading out the questions to the participants. Responses were recorded. No money was given to the participants but were acknowledged and appreciated for taking time out to participate in the evaluation.

3.1.6 .2 Participant General Instructions

Participants were asked verbally to use the think aloud technique to provide feedback of their interaction with the system. They were told that this will be recorded for analysis later. Participants were also told that they can interact with the test administrator by asking the test administrator any questions they encounter during the test.

3.1.6 .3 Participant Task Instructions

General task instructions for all the tasks required of by the participants were included with the tasks and tasks scenarios as listed in Appendix A. The following summarizes the tasks required of participants:

1. Browsing: Browse the Homepage and describe the structure, layout and the terminologies as you understand it.

2. Browse main clickable elements on the Homepage by clicking through and describing each sub-element and give a description of your understanding of structure, layout, and terminologies.

3. Perform typical course-support related tasks using Community@Brighton

3.1.6 .4 Usability Metrics

Measures and metrics of usability are indicators, both qualitative and quantitative, that informs system designers and developers about the usability status of a system. These measures drive design, development and re-design of systems to better meet the needs of the users of the system.
These measures are plentiful and diverse. These measures fall into three (3) main categories, namely effectiveness, efficiency and satisfaction with each category comprising a number of sub-metrics. The following section describes these metrics and their application in capturing usability issues in this study.

3.1.6 .5 Effectiveness

Effectiveness relates the goals of the system and the accuracy and completeness to which these goals are achieved. Frokjaer et al. (2000) [18] describes it as the 'accuracy and completeness with which users achieve certain goals'. They further suggested that typical indicators of effectiveness include 'quality of solutions and error rates'. Sharp et al (2006) [8] describes effectiveness as 'a very general goal' that 'refers to how good a product is at doing what it is supposed to do'.
In this study, completion rates, number of assists in solving a problem or completing a task and the number of defects are used as the primary indicators of effectiveness of the system.

3.1.6 .6 Completion Rate

Unassisted completion rate was defined as the percentage of participants who completed each hands-on based task correctly without help from the test administrator. Assisted completion rate was defined as the as the percentage of participants who completed each hands-on task completely with the help of the test administrator. Hands-on based tasks are listed in section three (3) of Appendix A.

3.1.6 .7 Defects

A defect was defined as a problem encountered with the system either through browsing the interface of the system or by carrying out the specific tasks. Some of these problems were related directly to the users' understanding of the system while others were as a result of the design and functionality of the system. Typical problems include: participants not understanding the terminologies used and systems' functionalities and lack of understanding of the structure and architecture of the system.
Defects were placed in on of the following four categories or sometimes as a combination where necessary:

1. Labeling
2. Functionality
3. Navigation
4. Heuristic – Using Nielson's 10 Usability [19]

Additionally, defects were not taken in totally but were classified using the number of occurrences of problems encountered by participants as indicated below:

- **a minor defect –** users made 2-3 attempts at explaining a concept of carrying out a task with eventual success
- **a major defect –** users took greater than 3 attempts at completing a task or explaining a concept but did so successfully, eventually after extensive trial and error.
- **total failure –** there was a complete failure at either explaining correctly a concept of completing a task.

3.1.6 .8 Assists

An assist was defined as verbal help given to the participant by the test administrator to guide the participant in the next step in completing a task. Participants were given at least 2 attempts at trying the task for themselves before an assist was given. There was no set limit on assists given unless time expired in the estimation of the test administrator. However, the number of assists was used to formulate the types of defects and problems encountered by the user.

3.1.6 .9 Efficiency

Frokjaer et al. (2000) [18] defines efficiency as the relationship between '(1) the accuracy and the completeness with which users achieve certain goals and (2) the resources expended in achieving them'. They further state that task completion time and learning time are indicators of efficiency. Sharp et al (2006) [8] provides a more flexible definition of efficiency by stating that 'efficiency refers to the way a product support users in carrying out their tasks'.

For this study, the problem count was used as a measure of the efficiency of the system for it is believed that there is a relationship, direct or indirect, between the numbers of problems encountered experienced using the system and the efficiency of the system. In fact, Lausen (2005) [9] supports this to an extent by suggesting that 'problem count is only and indirect factor that relates to efficiency

3.1.6 .10 Problem Counts or number of Errors

The number of defects or problems encountered per task by the participant was used as the only form or measure of the efficiency of the system. However, it should be noted that this is not often directly related to efficiency as pointed out by Lauesen (2005, p.28) [9] as 'experienced users from similar systems are likely to complain even if the user doesn't notice a problem'.

The number of major, minor or task failures per task were the value used to arrive at problem count. These measures represent to an extent, an indication of the amount of resources required to complete a task since a minor, major or task failure is defined in terms of the number of attempts at explaining an element or actions carried out in order to complete a task.

3.1.6 .11 Satisfaction

Frokjaer et al (2000) [18] describes satisfaction as the 'users comfort with and positive attitudes towards the use of the system'. Lauesen (2005) [9] defines satisfaction as simply 'how satisfied with the system'.

Perception information about users' satisfaction with the system was gathered using information solicited from task 1 and task 2 in the usability test and the subjective questionnaire at the end of the tasks. Additionally, satisfaction cues were gathered from users' comments during thinking aloud. Information about participants' perception on usefulness, appearance and ease of use were gathered.

3.1.6 .12 Understandability

Information about the participants' clarity and understandability of the system were solicited through the use of task 1 and task 2 and partly from the hands-on tasks of the usability test as listed in Appendix A. Information was solicited through the feedback provided by the users through thinking aloud and through their actions in seeking out a solutions to the tasks performed.

4.0 Data Analysis and Results

4.1 Analysis

4.1.1 Data Scoring: defects observed and experienced by participants behaviors was categorized into groups of minor defects, major defects and task failures. Behaviors were marked as a defect or an error when a participant undertakes an action that could not result in the completion of a task, when a participant could not correctly identify elements or predict outcome of an action. Likewise, each time the test administrator felt like a participant needed verbal assistance in order to complete a task, or to explain a concept or element, when the assistance was given for defects.
Specifically, 2-3 defects in carrying out a task but successfully completed the task, resulted in a task being rated as a task with 'minor defect', > 3 defects, but with success in carrying out a task resulted in a task defect being rated as a task with 'major defect'. A complete failure to complete a task, even with help from the test administrator resulted in a defect being rated as a task with complete failure or specifically, a 'task failure'.

4.1.2 Data Reduction: data for each task were analyzed individually and summarized together. Data was separated into three (3) main categories, with each representing holistically, one of the major tasks as indicated in the test in Appendix A. These categories were further, placed into four (4) groups namely, functionality, labeling, navigation and heuristic defects. More specifically, defects from within each of these 4 categories were grouped into three (3) categories namely: minor defects, major defects, task failure.

4.1.3 Data Analyses: descriptive statistics was used including totals and percentages. No inferential statistic was used to analyze performance.

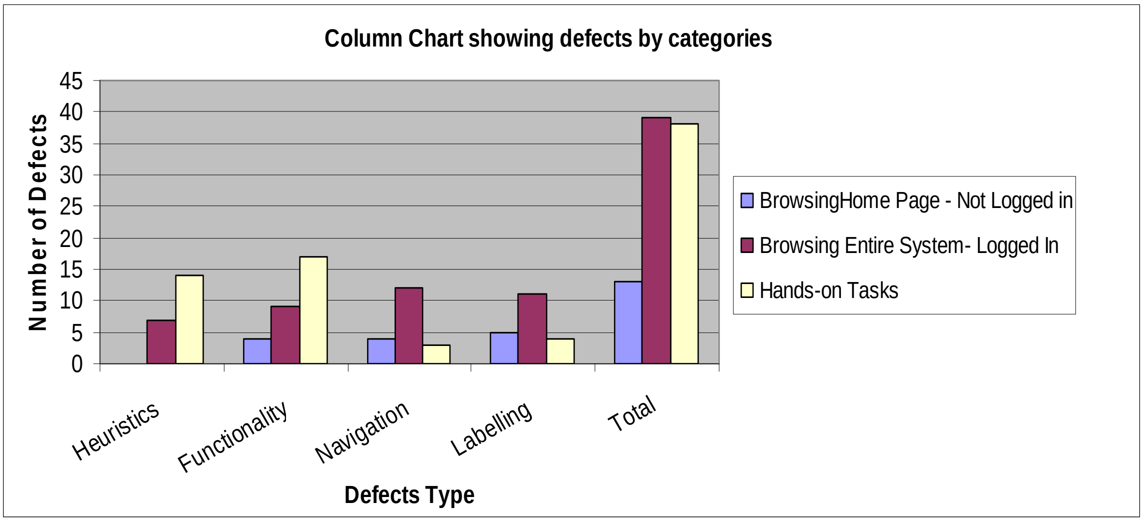
4.1.4 Results

Table [2] below provides a summary of all the defects discovered during the usability test.

Defects Type	**Sources of Defects and Count**			**Total**
	Browsing Home Page - Not Logged in	**Browsing Entire System – Logged In**	**Hands-on Tasks**	
Heuristics	0	7	14	21
Functionality	4	9	17	30
Navigation	4	12	3	19
Labeling	5	11	4	20
Total	**13**	**39**	**38**	**90**

A total of 52 defects were found for the browsing based tasks, 13 of which were discovered on the home page and the remaining 39 found while participants logged into the system and browse through the elements. A total of 38 defects were discovered during the hands-on based tasks. This resulted in a grand total of 90 defects.
Further, defects were placed into one of four categories for which there were 21 defects related to Nielson Usability Heuristics, 30 defects were identified as functionality based defects, and 19 were related to navigation and the remaining 20 related to the labeling and terminologies used by the system. Figure [y] below provides a graphical representation of this.

Figure [2] below shows the distribution of defects by location and categories



Each defect was categorized as either being a minor defect, a major defect or a defect that resulted in a complete task failure. The table, graph and pie chart below represents the distribution of these defects among the tasks carried out.

Table [3] below shows defects by location and defects type

Defects Severity	Sources of Defects and Count			Total
	Minor Problem	Major Problem	Task Failures	
Browsing Home Page - Not Logged in	8	5	0	**13**
Browsing Entire System - Logged In	10	22	8	**40**
Hands-on Tasks	7	9	21	**37**
Total	**25**	**36**	**29**	**90**

Figure [3] below shows the distribution of defects by Tasks and location

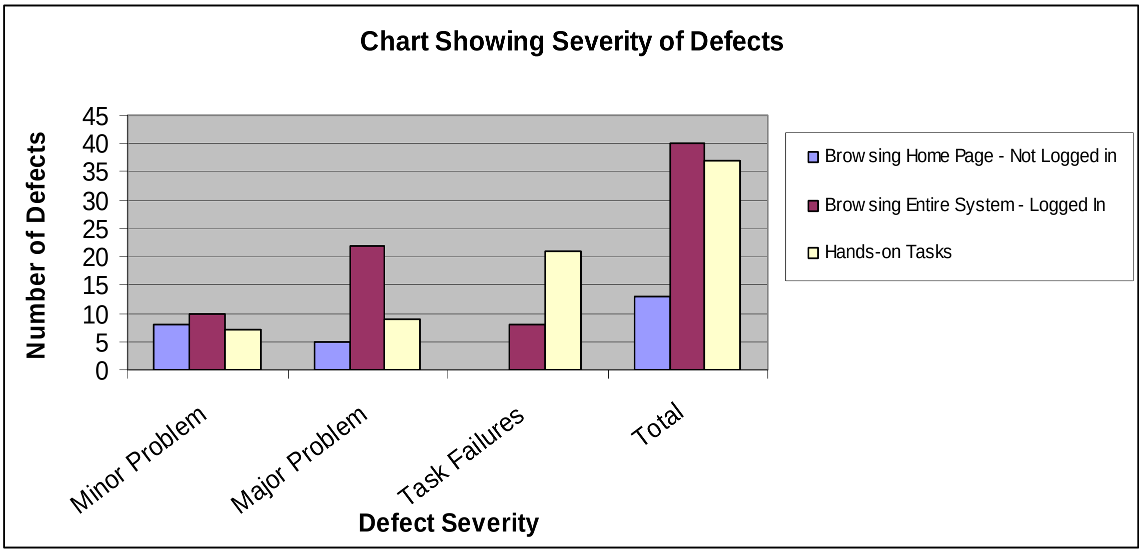
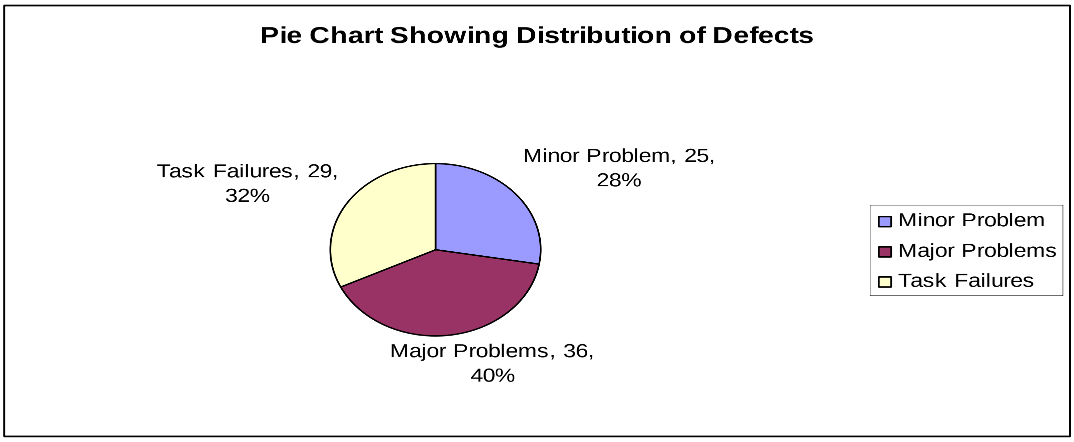


Figure [4] below shows the distribution of defects by Tasks and location



Performance Results of Hands-on Tasks

Task	Participants				
	RG	**ME**	**JW**	**RH**	**DH**
1	CC/WA	CC/WA	CC/WA	CC/WA	CC/WA
2	CC/WA	CC/WA	CC/WA	CC/WA	CC/WA
3	**Cin**	**CC/Wi**	**CC/Wi**	CC/WA	**CC/Wi**
4	CC/WA	CC/WA	CC/WA	CC/WA	**CC/Wi**
5	**Cin**	CC/WA	**CC/Wi**	CC/WA	**CC/Wi**
6	**Cin**	CC/WA	CC/WA	CC/WA	**Did not attempt**
7	**Cin**	**Cin**	CC/WA	CC/WA	**Did not attempt**

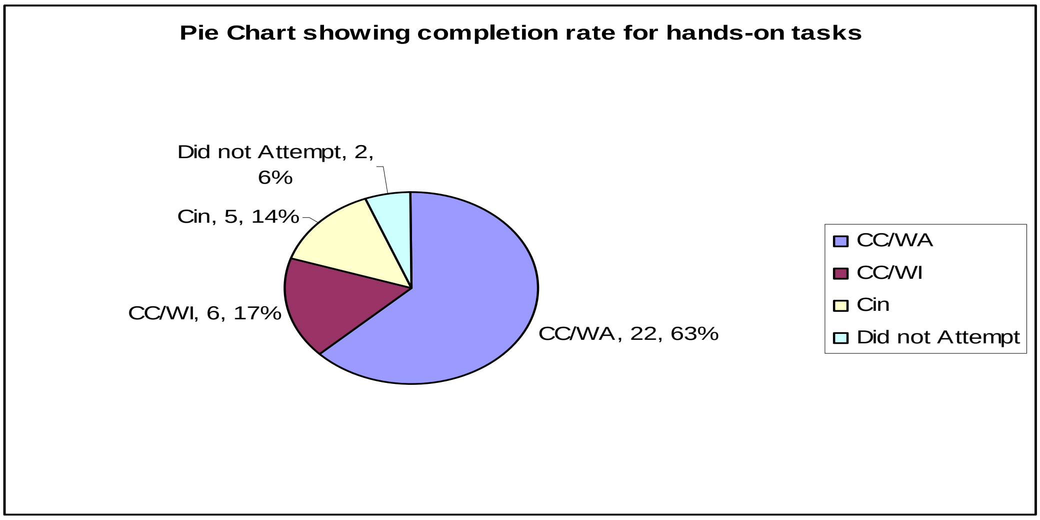
CC/WA - Completed Correctly With Assistance

Cin – Completed Incorrectly

CC/Wi - Completed Correctly Without Assistance

Table [4] above shows performance results of participants on the hands-on based tasks.
All of the participants completed each of the seven (7) tasks required, with the exception of one participant who did not attempt of the tasks due to tiredness. Of the five (5) participants, none completed all the tasks correctly on their own. One participant completed all the tasks correctly but with assists from the test administrator for each task. Another completed all the tasks but needed assistance for five of the tasks. One participant completed all the tasks but did 4 without assistance from the test administrator, incorrectly.
Tasks 1 and 2 were the required complete assistance but were all completed correctly. Tasks 3, 4 and 5 were the most successfully completed tasks in the sense that 3 of the 5 participants completed the tasks on their own i.e. without assistance. 17% of the tasks were completed correctly without any assistance. 63% of the tasks were completed correctly with assistance while 14% of the tasks completed were done incorrectly. 6% of the tasks were not attempted.

Figure [5] below shows completion rate of hands on tasks

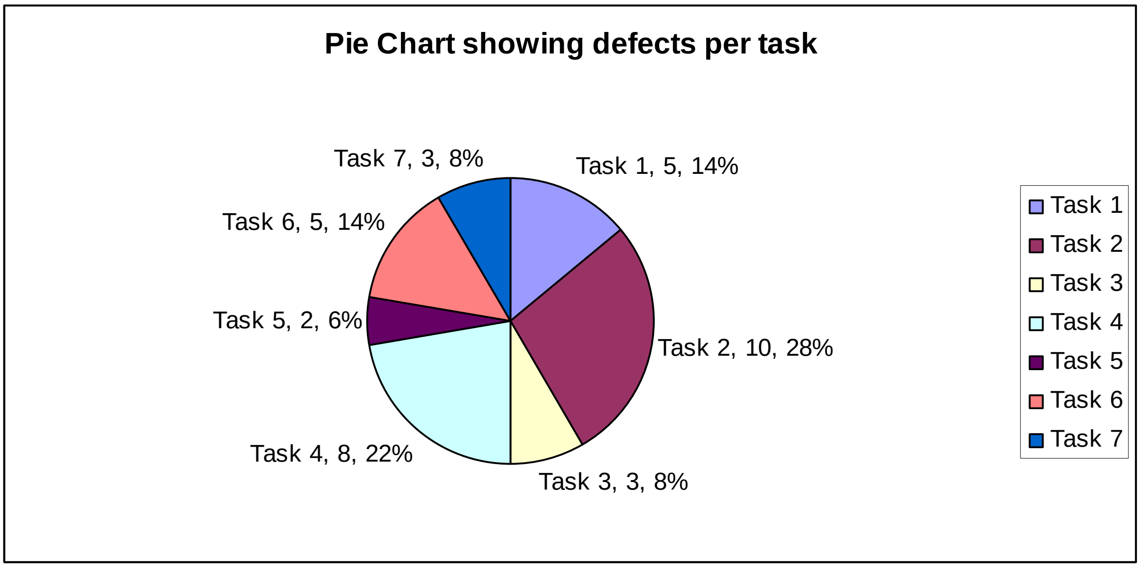


Defects for these seven tasks, grouped by minor, major or task failures are presented in the table [5] below:

	Defects Severity			
Task	**Mi**	**Ma**	**TF**	**Total Defects per Task**
1	1	0	4	**5**
2	3	3	4	**10**
3	2	0	1	**3**
4	1	5	2	**8**
5	0	0	2	**2**
6	0	0	5	**5**
7	0	0	3	**3**
Total	**7**	**8**	**21**	**36**

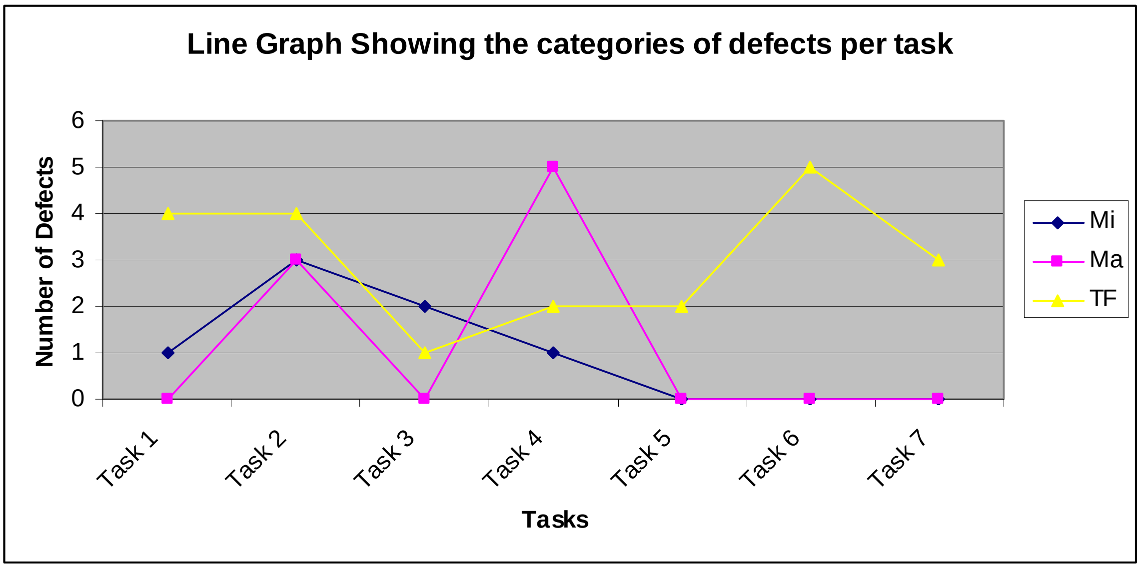
Of the 36 defects discovered, 21 or 58% relates to complete task failures while 22% and 20% relates to major and minor tasks defects respectively. The pie chart below gives presents a breakdown of these defects by task.

Figure [6] below shows defects per tasks in hands-on activities



The line graph below provides a further breakdown of the number of defects per task by showing the number of defects per category for each task.

Figure [7] shows defects count per tasks in hands-on activities



4.2.0 Summary of Satisfaction Results

Test participants were asked to provide feedback on their satisfaction using the system using the open-ended questionnaire in Appendix B. This questionnaire was administered by the test administrator and participants' verbal responses were recorded using Morae.

There were mixed reactions from participants' general satisfaction with the system. 3 of the 5 participants would recommended the system but for experimental purposes and not for immediate use for course support. A fourth participant completely rejected the system, citing it as being too 'clunky', while the other participant saw 'no obvious advantage' in using the system. However, all of the participants provided positive feedback about the kinds of features and functionalities provided by the system.

Participants questioned the relative emptiness of the system and the reasons for it for e.g. there were many blogs and communities with no content or activity. They were generally suspicious about the motives behind the development and future use of the system.

Participants' satisfaction with the systems seems to have been influenced also by their prior experience using another system for course support – StudentCentral [21]. Participants consistently compared the two systems and the ease with which StudentCentral [21] allowed them to accomplish similar tasks. Specifically, participants noted that StudentCentral [21] allows for simple things to be accomplished easily even for the novice user even though they acknowledged having difficulties using some of its features.

5.0 Discussion of Results, Recommendation and Future Work

There were a number of issues related to the syntax and semantics of the system that proved problematic for participants to understand. Many of the issues on syntax were related to participants understanding of terminologies used by the system while semantic issues were associated with some system functionalities.

Specific problems were noted when participants tried to verbally describe concepts such a 'Tag Cloud', FOAF – Friend of a Friend, 'Friends Request' etc. Clearly, these are terms commonly associated with social software systems but participants struggled to explain correctly what these terms were. Also, there was a great struggle in understanding how the system under these terms operated.

Three (3) observed semantic issues that participants found difficult to come to terms with are related to the use of 'tags', the search functionality of the system, and the concept of a personal space and a community space, being separate logical entities in their own rights but physically existing in the same space.

Participants expected the search to return results presented by topics or titles search for. However, they were presented with search results by people, communities and tags. This proved a difficult concept for participants to understand even though they would have appreciated the idea.
Participants appreciated the concept of tagging things but they did not understand how tags were generated. This is probably best reflected in their constant reluctance to supply tags information during their interaction with the system.
There was great difficulties experience by participants in using their own space for personal use and the community space for community matters. This confused and unclear state of being is compounded by the fact that a participant can be member of many communities and at the same time be an owner of communities. What was very evident was the consistency at which participants' added content to their personal space instead of the community space. 4 of the 5 participants repeatedly did this. However, there were signs of participants being unsure about the accuracy of what they did.
The problems discovered during usability testing are most useful for informing the re-design and development of the user interface of a system. Solving all the problems and fixing usability problems would be impossible and therefore the need to prioritize usability problems.
Dumas & Redish (1999) [15] suggests that the 'organizing principle that makes sense is importance, which they define in terms of what they called the scope and severity dimensions. The define scope in terms of 'how widespread the problem is' and severity as 'how critical is the problem'. Further, Nielson [20] defines severity as a combination of three (3) factors namely: frequency of the problem, the impact of the problem, and the persistence of the problem.
Using these underlying theoretical foundations as guiding principles, a prioritized list, providing suggestions for fixing usability problems for each of the four categories of defect, is proposed. It takes into consideration the results for severity levels of the defects together with the test administrators own intuitive judgments about the system. This guideline should not be taken as an absolute or complete since the test administrator is not part of the systems design and development. They are meant to provide insights and ideas about possible solutions to the problems discovered. Defects are presented in order for which they are proposed to be fixed as shown below.

6.0 Limitations of Study

Participants for this study were drawn from the population of lecturers within the School of Computing, Mathematical and Information Sciences. An available sample was used i.e. the 5 participants were those available during the time of this study and not a random sample. Participants were very inexperienced with using the system under study and were almost completely new to social software systems such as the one investigated. Background knowledge about the participants will most definitely lend it self to a number of questions that one could ask pertaining to the results of this study.
The question about the minimum number of participants to use in a usability study for effective results is a well documented one. Five (5) participants were used for this study. This is a well noted number in the field of usability evaluation and its use was justified by that.
How suitable was this group of participants given that they were evaluating a system that is part of the University of Brighton and therefore, the participants' set of available tools, for use? Additionally, participants' were asked to undertake tasks similar to those they are accustomed to, using another system at the University of Brighton i.e. StudentCentral [21]. It is most interesting to note that one (1) of the five (5) participants is a specialist in interface design and evaluation methodologies and this may have affected his approach to the evaluation.
Would another group of similar participants, not associated with University of Brighton, provide more authentic results? Would using a set of students as participants for the study provide different results given that students are likely to be more experienced with using social software system than lecturers? This is an interesting issue that calls for further investigation and discussion since it may very well prove to be an important factor in evaluating the results of the study.

7.0 Conclusion

This paper presented the findings of a study conducted as a preliminary effort to identify potential usability issues associated with a social software system used in a University environment. This study was done using a traditional approach to usability evaluation – laboratory based usability testing.

A number of issues were identified with the system using this approach and results suggest that there might be some usability issues with this social software system and social software systems in general. However, whether these issues are really of great concern needs to be treated with some caution for a number of reasons. A limited number of participants were used in the study and they are from a similar demographic group. Additionally, it is not yet known with any degree of certainty whether traditional approaches to studying the usability of social software systems will identify issues with this new form of interactive social system.

Further research is needed to extensively identify issues associated with such systems, perhaps using other approaches to usability evaluation with a more diverse range of participants. It is the purpose of this preliminary study to shed some light on potential usability problem areas with social software systems using traditional methodologies and to further debate the issue of usability in social software systems.

Appendices

Appendix A

Elgg – Community@Brighton - Evaluation Tasks

Interface Tour – Browsing-based Tasks

1. Open the home page of the site.

1. Have you ever seen this Web site before?
2. How many times approximately?
3. What was the purpose for your visit(s)?
4. Please give me your initial impressions about the layout of this page and what you think of the layout, colors, graphics, photos, etc.
5. Without clicking on anything yet, please describe the options you see on the home page and what you think they do. Feel free to move around the page, but again I will ask you not to click on anything right now.
6. Without clicking on anything yet, if you were exploring, what would you click on first?
7. Why did you decide to click that first?
8. What do you think is the purpose of this site?
9. Who do you think this site is intended for?
10. Who do you think is responsible for this site?
11. Who do you think is expecting you to use this site? Who will know if you do/don't use it? Do you think you would miss any information if you don't use it?

2. You are required to go through/click through the interface one (1) element at a time and in a few words talk about each element using the following guidelines:

- Briefly describe each element
- State whether the names of each element make sense
- Where you expect each element to take you when you click on it? What kind of page you expect to find on the other side? What would it contain? How would it look?
- What is the most important thing on each screen for you?
- Is there any information missing from here that you would need?
- How would you get to the front door of this site from each page? What would you click on?

3. Task-based Activities

1. You have just recently taken over the Module SWM 35 – Web Mastery and are thinking about creating an online community for the students of this course. Create a community called SWM 35 – Web Mastery 2007 for this Module.
2. You have created your first lecture using Microsoft PowerPoint (Lecture1.ppt) and would like to share this with your SWM 35 class. Upload this lecture to the students of the SWM 35 Community in a folder called SWM35_Lecture1.
3. Your students have all received the first lecture (Lecture1.ppt) from the community online from which you expect them to clarify any initial issues they may have experienced after going through this Lecture. Set up a Blog Post called 'Lecture1 – Feedback Blog', asking your students to provide their comments, feedback and questions using this blog.
4. After a few days your students have provided some valuable feedback on lecture one in the Lecture1-Feedback Blog, including comments and questions for which they expect you to provide answers and further guidance. You have found some additional resources that you believe may help them answer some of their questions. The resources you want to pass on to them include two (2) URLs for online resources. Create a follow-up Presentation called Lecture1_Help and add a section that allows you to provide these URLs to your students. What other features do 'things' do you expect this Presentation can help you do?
5. You believe that there might be other communities sharing similar interests to some of the discussions in SWM 35, which might be useful for your students. Identify at least two (2) such communities and make them available to your students.
6. You have found an interesting website with excellent content which you think might be useful for SWM 35. This website provides an RSS feed that can be embedded into your website. Include this RSS feed into the SWM 35 Community.
7. You would like to know if there are discussions or topics similar to those taking place in SWM 35 on this website. How would you go about finding out about these topics of discussions?

4. If you were asked to explain this system to a colleague or student, how would you represent your explanation diagrammatically to them?

Appendix B

Follow-up Session – Wrap-up & Brain Storm

1. Is this an interesting application? Is this something that you would use to support your module?
2. How would you describe this product in a couple of sentences to someone with a level of computer and web experience similar to yours?
3. Is this something you would recommend? Why/Why Not?
4. By summarizing what we have looked at so far, can you say 2 good things and 2 bad things about this application?
5. What would you like a system like this to do that this one doesn't? Have you ever felt like saying 'I wish this system could have done X or Y for me'? What would it be?
6. Do you have any final questions? Comments?
7. Feel free to email us with any other thoughts or ideas on your way home, tomorrow on over the next week. (lenandlar@yahoo.com)

References

[1] http://community.brighton.ac.uk
[2] **Koskinen, T. 2006.** Social Software for Industrial Interaction. Proceedings of the 20th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artefacts and environments OZCHI '06. ACM Press.
[3] http://www.elgg.org
[4] http://elgg.org:81/mod/mediawiki/wiki/index.php/License
[5] **Preece, J. (1993).** A Guide to Usability: Human Factors in Computing. Addison Wesley. Wokingham, UK.
[6] http://cost294.org/uess2007/
[7] **Nielson, J., 1998.** The Design of Everyday Things. MIT Press. London
[8] **Sharp, H., Rogers, Y. Preece, J.,** 2006. Interaction Design: beyond human-computer interaction. John Wiley & Sons Ltd, West Sussex, England.
[9] **Lauesen, S., 2005.** User Interface Design: A Software Engineering Perspective. Pearson Education. Essex, England.
[10] **Chan, H.C., Teo, H.H.** 2007. Evaluating the Boundary Conditions of the Technology Acceptance Model: an exploratory investigation. ACM Transactions on Computer-Human Interaction (TOCHI), Vol 14, Issue 2. ACM Press.
[11] **Nielsen, J.,** 2000. Why you only need to test with 5 users. Alertbox (2000) http://www.useit.com/alertbox/20000319.html
[12] **Virzi, R.A.1992.** Refining the test phase of usability evaluation: How many subjects is enough? Human Factors, 34, HFES (1992), 457-46
[13] **Nielsen, J., & Landauer, T. K.** 1993. A mathematical model of the finding of usability problems. In Proceedings of INTERCHI 1993, ACM Press (1993), 206-213.
[14] **Lindgaard, G., Chattratichart, J. 2007.** Usability Testing: What have we overlooked? Proceedings of the SIGCHI conference on Human factors in computing systems CHI '07. ACM Press
[15] **Dumas, J.S., Redish, J.C.** 1999. A Practical guide to usability testing. Intellect Books. Exeter, United Kingdom.
[16] http://www.cmis.brighton.ac.uk/staff/jd29/
[17] http://www.techsmith.com/morae.asp#
[18] **Frokjaer, E., Hertzum,M., Hornbaek, K.** 2000. Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated? Proceedings of the SIGCHI conference on Human Factors in Computing Systems, CHI 00. ACM Press.
[19] http://www.useit.com/papers/heuristic/heuristic_list.html
[20] http://www.useit.com/papers/heuristic/severityrating.html
[21] https://studentcentral.brighton.ac.uk/webapps/login/

Article received: 2009-11-22