Proposal of a User Interface Design Guideline based on Analysis of the Causes of Usability Defects reported in a project-based Software Engineering Course

Takuro Minoya[1], Kiichi Furukawa[2] and Atsuo Hazeyama[1]

[1] Tokyo Gakugei University, 4-1-1 Nukuikita-machi, Koganei, 184-8501, Japan
[2] Graduate School of Education, Tokyo Gakugei University, 4-1-1 Nukuikita-machi, Koganei, 184-8501, Japan

Abstract

In project-based learning (PBL) for software development (PBL4SD) conducted in the 2020 academic year at Tokyo Gakugei University, the development task was to develop a web application for use in schools. In the acceptance testing phase, we confirmed 2.5-times as many defects as those of the previous year. We analyzed the trends and characteristics of these defects and found that many were related to usability. In addition, there was a case in which the changes to the inspection comments were reflected in the artifacts created by the user interface (UI) design, but not in the system, and were reported as defects in the acceptance testing phase. To solve these problems, we propose a UI design guideline for PBL4SD.

Keywords

Software engineering education, software development, project-based learning (PBL), usability defects, and usability design guideline

1. Introduction

In recent years, project-based learning (PBL) has received attention as an educational method promoting active learning by learners. Along with this trend, PBL for software development (hereafter, PBL4SD) has been actively conducted in the field of informatics education. Kumeno et al. reported that PBL-based exercises brought about changes in the skill acquisition of learners, improvements in the learning effectiveness, and an increased learning motivation of the learners [1]. Based on our experience regarding the long-term operations of a PBL4SD course, we agree with their claims.

Although most university students in computer science departments study programming in their first year, they do not have as much software development experience in comparison. Although software engineering textbooks include a number of theoretical and conceptual themes, students cannot obtain as much experience simply by reading such textbooks as they can through actual software development. Therefore, practice is an important aspect.

In particular, this study is focused on usability defects. According to ISO 9241-210, usability is defined as the degree of effectiveness, efficiency, and user satisfaction when a product is applied to attain specified goals by specified users under specified usage situations. We define usability defects as anomalies in terms of the degree of effectiveness, efficiency, and user satisfaction.

In our experience, despite conducting a user interface (UI) design inspection, defects are frequently reported during acceptance testing, which is the final phase of the course. This is a valuable learning experience because students learn through real software development. However, because the workload of the students has increased, a significant effort may lead to unfinished projects. We would therefore like to reduce the amount of backtracking. It is necessary to guide the learners, and our approach to reducing backtracking is to enhance the quality during the upstream phase, that is, the UI design phase. We therefore focused on improving the UI design.

Studies on usability defects and the proposal of guidelines for awareness regarding usability

Proceedings of 6th Software Engineering Education Workshop (SEED 2022) co-located with APSEC 2022, 06-Dec, 2022, Japan (Virtual)
EMAIL: hazeyama@u-gakugei.ac.jp (A. 3)
ORCID: 0000-0002-6583-1521 (A. 3)

development have previously been conducted [2]–[4]. To the best of our knowledge, however, studies on usability defects targeted at PBL4SD and those attempting to decrease the number of such defects have yet to be conducted.

This paper is focused on usability defects that occur in PBL4SD, and UI design guidelines are proposed.

The rest of this paper is organized as follows. Section 2 describes the work related to this study. Section 3 describes the investigation of this practice. Section 4 presents the results of the investigation. Section 5 presents the discussions of this investigation. Section 6 proposes a UI design guideline based on the discussion in section 5. Section 7 shows a preliminary evaluation of the proposed UI design guideline. Finally, section 8 summarizes the study.

2. Related work

In this section, we provide a state-of-the-art analysis of usability defects and guidelines for a usability design. We then discuss the position taken by our study.

2.1. State-of-the-art of analysis of usability defects

Yusop et al. conducted a systematic literature review (SLR) to ascertain a state-of-the-art reporting and analysis of usability defects and key challenges [5]. The authors made several recommendations to improve the usability, defect reporting, and management in software engineering.

2.2. Systematic rules for usability aware design

Shneiderman presented eight rules for UI design. Based on Shneiderman’s rules and experiences, Nielsen proposed 10 general principles for interaction design. Lohdi assessed Nielsen’s principles as parameters for conducting usability testing and showed that such principles make good parameters [6].

2.3. Position of our study

This study investigates the usability defects in PBL4SD by associating them with Nielsen’s principles. We then propose a guideline for a user interface design for novice learners.

3. Investigation

This section provides an overview of the target PBL4SD course and project. Next, we present a data-extraction method for UI design inspection and acceptance testing. We then set up the research questions.

3.1. Overview of the target PBL4SD

The PBL4SD course is offered to third-year undergraduate students in the Department of Informatics Education at Tokyo Gakugei University. The department quota is 15 students. Therefore, the PBL course is conducted on a small scale. The team is organized into three to five students. The course consists of 15 weekly 90-min long lectures. The task given to the students is web application development using Java. In the preceding semester, we provide an introductory software engineering course.

Among other aspects of PBL, regarding the application of the course, in the information provided to the students, we specify the software development process, artifacts, and approach to grading and provide an in-depth explanation during the first lecture.

The development process is based on the waterfall model. The types of artifacts that each team is required to create are requirements specification, a user interface design document, a class diagram, a database design document, sequence diagrams, source codes, unit/system testing reports, development plan, team progress reports (each week), and a project completion report. A sequence diagram and source code are created, and unit testing is conducted for each function by each student.

Verification activities, that is, software inspection and testing, are conducted. Software inspection is conducted by the teaching staff (teaching assistants and the teacher) for artifacts of the requirements specification, UML diagrams, and a database design document. After system testing by the teams, acceptance testing is conducted by the teaching staff.

The team progress report is presented in turn by each team member. To provide feedback to the student teams, progress checks during the lecture time, the inspection of artifacts created during the upstream phase, and acceptance testing of the application developed by each team are conducted by the teaching staff.

As the development environment, students use their own laptop computers. We use GitHub as a source code and document repository. In addition, we use the version control in the documents, the "Issues" function (the formal location of our text communications, including discussions and defect reporting, among other exchanges), and the "Pull Request" function for the artifact review process. The teams are allowed to use various tools.

3.1.1. UI design

By referring to the requirements specification, each team creates a user interface design document, which is composed of screen images for each function unit provided by the system and the transitions among the screens.

3.1.2. Acceptance testing

The teaching staff in charge of customers conduct acceptance testing for the released system. If problems are detected, they write issue reports and submit them to GitHub.

Problems with the system are reported using the issue function on GitHub. The team checks the reported issue reports, and if they are judged as defects, the developer who implemented the function modifies the source code to fix the defects and returns the issue reports. At this time, the developer is required to describe the lessons learned from this activity.

3.2. Target PBL

In the 2020 academic year, nine students took the course and were organized into two teams. One team was made up of four students and the other by five students. The task was to create a menu application for elementary and secondary school students.

3.3. A method to extract inspection comments from UI design inspection

We need to extract issues regarding the UI design inspection from all issues in the repository for the 2020 course. We prepared label "inspection" and set a rule that this label should be attached for inspection-related issues.

3.4. A method to extract defects data from acceptance testing

We need to extract issues regarding the usability defects found during acceptance testing from all issues within the repository for the 2020 course.

We prepared a "defect" label and set a rule stating that this label should be attached to the issue reports provided during the system and acceptance testing.

We judge those issues that were reported by the teaching staff or clients during acceptance testing, and words such as "usability," "difficult to use," "difficult to see," and "operations are difficult" were included in the title and/or body of issues as usability defects. The defects were counted for each issue.

The followings are excluded for analysis:

- issue duplicated defects
- an issue was reported as a "defect" but was closed as "not a defect" when the result of the investigation was excluded.

3.5. Research question

We next present research questions for proposing the UI design process from our practice.

RQ1: What is the ratio of usability defects for all reported defects in acceptance testing?

RQ2: What is the relationship between the usability defects and Nielsen's principles?

RQ3: What is the relationship between inspection comments regarding the UI design document and defects reported in the acceptance testing?

4. Result

We answer each research question based on the results of the analysis.

4.1. Answer to RQ1

A total of 86 defects were identified in the 2020 course. Among them, 40 were usability defects (46%).

4.2. Answer to RQ2

Table 1 shows the relationship between usability defects and Nielsen's 10 principles. As the reason why the total number exceeds 40, a defect is related to multiple principles.

Table 1 shows that most of the defects were related to No. 4 “Consistency and standards” Among them, six were problems with inconsistency in the size and placement of objects. For example, although some “return” buttons exist in the system, their sizes and locations are inconsistent in the system. There are three problems with inconsistencies in the terminology. For example, the forms for entering a password differ page by page. One page shows “enter password,” whereas another page shows the length of characters of the password “within X characters.” There is one inconsistency problem among the entire design.

The second most common defects were related to No. 8 “Aesthetic and minimalist design.” “Unnecessary explanations on a page confuse the users” and “necessary information is lacking” were pointed out as defects.

Table 1
The relationship between the usability defects and Nielsen’s principles

No	Title of the principle	No. defects
1	Visibility of system status	3
2	Match between system and the real world	3
3	User control and freedom	3
4	Consistency and standards	10
5	Error prevention	5
6	Recognition rather than recall	4
7	Flexibility and efficiency of use	5
8	Aesthetic and minimalist design	7
9	Help users recognize, diagnose, and recover from errors	2
10	Help and documentation	3

4.3. Answer to RQ3

We present some characteristics of UI design inspection comments and describe the relationships between comments and defects during acceptance testing.

4.3.1. Comments pointed out in UI design inspection

Inspection comments from the perspective of functionality and UI were found in both teams.

Regarding functionality, elements written in the requirements specification are missing or ambiguous in the UI design document. For example, although the logout function is written in the requirements specification, the corresponding page and/or buttons are not found in the UI design document.

For the UI, the flow of operations used to conduct a function is difficult for users. In addition, inconsistencies in the design, such as color, size, and layout of the buttons and/or characters, are pointed out. Furthermore, as application-specific comments, the usage of Chinese and Japanese characters under a particular situation is pointed out because this system is intended to be used by elementary school students.

4.3.2. The relationships between inspection comments and defects in acceptance testing

From Table 1, defects regarding principles 4 and 8 by Nielsen were reported more during the acceptance testing in our case study. Seven of the 17 defects reported as items 4 and 8 were also pointed out in the UI inspection. Although the comments pointed out in the UI design were reflected in the UI design document, they were not reflected in the system, and they were reported as defects during acceptance testing.

5. Discussion

We provide discussions from the results of the investigation to the 2020 PBL4SD course.

5.1. Some defects can fix at the UI design phase

As a result of RQ3, usability defects were identified in the UI design inspection. Inspection comments for the UI design document are not reflected in the system but are reflected in the revised UI design document. The UI design document should be reflected in the system.

5.2. Students tended to develop a system without considering usability

until they pointed out usability defects

The ratio of usability defects to all defects in the acceptance testing was approximately 46%.

We found some reflection comments in the issue report such that “I was not aware of the usability” and “I did not develop from the viewpoint of a user.” We found that the developers had a low recognition of usability under development, but with awareness of the usability from usability defect reporting during the acceptance testing.

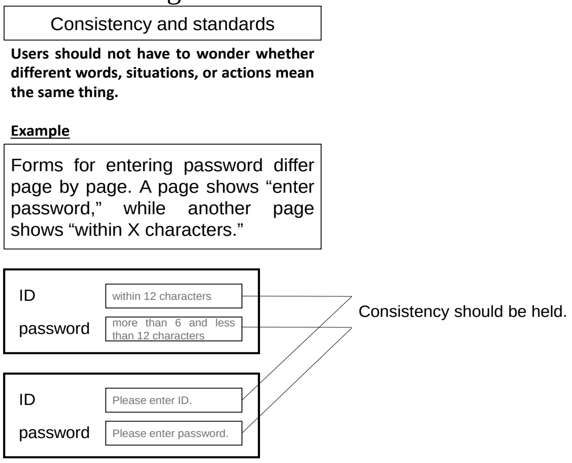
We believe that supporting a usability-aware design and implementation will lead to a decrease in usability defects during acceptance testing.

6. Proposal of a UI design guideline

We propose a usability improvement method for use in UI design based on the abovementioned discussion.

From the results of RQ3, we found a relationship between comments regarding the UI design inspection and usability defects reported during acceptance testing. To decrease the number of usability defects, developers should have concrete awareness of the usability inherent to the design. We propose a UI design guideline that associates Nielsen’s 10 principles with the real usability defects that have occurred in our course.

Figure 1 shows an excerpt of the proposed guideline. For each principle, we prepared an explanation written by Nielsen and examples detected during our course.



Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing.

Example

Forms for entering password differ page by page. A page shows “enter password,” while another page shows “within X characters.”

Figure 1: Proposed usability guideline (excerpt)

7. Preliminary evaluation

In this section, the results obtained from applying the developed usability guideline to an actual PBL4SD course are reported.

7.1. Overview of the evaluation

The usability guideline was applied to the 2021 course. One team, made up of four students, was organized during the school year. The assigned task was the development of a school affairs support tool that allows a teacher to record all student assessments on a user interface, similar to an actual classroom (arrangement of desks). The development process, artifacts to be created, and the software platform used (Microsoft Teams and GitHub) were all the same as those applied during the 2020 academic year.

We delivered the usability guideline when the team started designing the user interface. We also made an announcement confirming the usability guideline and the comments from the inspection of the user interface design document at the start of the implementation.

We set up two research questions to confirm the effectiveness of the proposed usability guideline:

RQ4: We clarify the ratio and characteristics of the usability defects reported during the 2021 PBL4SD course.

RQ5: Based on a questionnaire, we identified the merits and improvements of the usability guideline.

7.2. Result

7.2.1. Answer to RQ4

A total of 41 defects were reported during the acceptance testing. Among them, 18 were usability defects (the ratio was approximately 44%). The ratio was almost the same during the previous year. The introduction of the usability guideline did not always decrease the ratio of usability defects. At most, four defects regarding the “visibility of the system status” were reported, whereas defects regarding the “consistency and standards” and “aesthetic and minimalist design,” which were the highest in 2020, decreased to three and zero in 2021, respectively.

7.2.2. Answer to RQ5

We collected descriptions on the merits of the proposed usability guideline from the students through a questionnaire.

- I came to understand what should be noted in a user-friendly interface design and practiced developing such aspects.
- I can imagine the usability defects that occur in the development of an application in a concrete manner.
- In designing a user interface, I can consider how the users can feel at ease by considering where the buttons are to be placed and what colors should be used.
- I can confirm the important points in a user-friendly interface design in advance.

All students answered that they recognized no demerits regarding the usability guideline.

By contrast, the following improvements were presented:

- Usability defects that do not appear in the usability guideline were reported.
- Examples of other systems are helpful to me.
- The guideline provides a large amount of detailed information (with a large number of pages).

The current usability guideline lacks comprehensiveness because usability defects were reported during the 2021 acceptance testing, which applied the usability guideline. In addition, because some other examples were requested, the usability guideline needs to be enhanced.

The team conducted an elaborate user interface design based on the usability guideline, particularly the “help and documentation” item. This resulted in an increased burden. It is necessary to consider a usability design that considers the documentation without drastically increasing the number of screens composing an application.

8. Summary

To reduce the number of usability defects in acceptance testing of PBL4SD, we analyzed the defects and found that they can be decreased by improving the UI design phase. We then proposed a UI design guideline that associates Nielsen’s 10 principles with the real usability defects that occurred during our course. We applied the proposed guideline to an actual course and evaluated it. While the learners responded effectiveness of the guideline, some improvements are identified.

We will improve the guideline based on the improvement comments and continue to apply it to the future course.

9. Acknowledgements

This study was partially supported by the Grant-in Aid for No. (C) 20K12089 from the Ministry of Education, Science, Sports, and Culture of Japan. We thank the students who took the course.

10. References

[1] Fumihiro Kumeno, Yasuhiro Tsujimura, Mikio Ohki, and Hidemi Yamachi, Software development PBL for solving real problems in local communities, IPSJ Transaction on Education and Computer, Vol. 2, No. 1, pp. 25-40, 2016 (In Japanese).

[2] Ebba Thora Hvannberg and Effie Lai-Chong Law, Classification of Usability Problems (CUP) Scheme, In Interact, Citeseer, 2003.

[3] Natalia Juristo, Ana M Moreno and Maria-Isabel Sanchez-Segura, Guidelines for eliciting usability functionalities, IEEE Transactions on Software Engineering, Vol. 33, No. 11, pp. 744-758, 2007.

[4] Laura Carvajal, Ana M Moreno, Maria-Isabel Sanchez-Segura and Ahmed Seffah, Usability through software design, IEEE Transactions on Software Engineering, Vol. 39, No. 11, pp. 1582-1596, 2013.

[5] Nor Shahida Mohamad Yusop, John Grundy and Rajesh Vasa, Reporting usability defects: a systematic literature review, IEEE Transactions on Software Engineering, Vol. 43, No. 9, pp. 848-867, 2016.

[6] Afifa Lodhi, Usability Heuristics as an assessment parameter: For performing Usability Testing, Proceedings of the 2nd International Conference on Software Technology and Engineering, Vol. 2, pp. 256-259, IEEE, 2010.