ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

Research on Online and Offline Teaching of Software Operation Courses

Wei Zhang, Ying Li, Dong Wang*, Fei Tang, Jiarong Li, Hui Li

School of Art and Design, Southwest Forestry University, SWFU Kunming, China.
577129801@qq.com, 1593370809@qq.com, 362511597@qq.com, 18701416441@163.com, 1205755171@qq.com, 984247350@qq.com
**Corresponding author: Dong Wang*

Abstract: *In order to promote the reform of online and offline hybrid teaching, improve the teaching quality of software operation courses in universities, and help students improve the learning effect of software courses. In this study, a hybrid research method of quantitative and qualitative was used to carry out an empirical study by taking sophomores, juniors and seniors from the College of Art and Design of Southwest Forestry University as examples. Questionnaires were used to collect students' proficiency in the software of this major. Based on the analysis results of the questionnaire survey data, the students who have mastered more than 75% and less than 25% of the software of their major were selected for the group interview, and the main influencing factors of students' good or bad mastery of software were explored through qualitative analysis method. Based on the findings of the study, about 52% of the students rated their software proficiency as average, 29% were satisfied (above 75%), and 56% were dissatisfied (below 25%). The main factors that affect students' mastery of software are that they don't practice enough; Poor absorption of classroom knowledge and the school curriculum arrangement is unreasonable.*

Keywords: *Software operation courses; Mixed teaching; Online and Offline*

1. Introduction

From the perspective of teaching mode, most of the software operation courses in universities in China still mainly adopt the traditional offline classroom teaching mode, for that reason there is a lot of room for improvement in the teaching quality. With the rapid development of new media, it has become an urgent momentum to explore a glorious avenue of software curriculum education in line with the educational concept of the new era. The coming of the era of "Internet + Teaching" has a profound impact on the reform and research of the online and offline hybrid teaching of software operation courses in universities in China. The mixed online and offline teaching mode is mainly embodied in two aspects: The first aspect is face-to-face offline teaching, in which teachers of our school will teach basic knowledge and solve "difficult and miscellaneous problems" in learning. The second aspect is online courses, which are arranged by the school and selected by students themselves, so as to familiarize themselves with relevant knowledge in advance and consolidate knowledge after class. These include: watching teaching videos, reading documents and related knowledge, and completing knowledge tests and group learning tasks. Students are directly involved in the course teaching and have a close relationship with teaching.

In the teaching of software operation courses, how to let students keep up with the pace of The Times and design works in line with the requirements of The Times? There are three points. Firstly, innovative teaching equipment; Secondly, constantly update the teaching content; Thirdly, the teaching methods should be improved. So the application and promotion of the online and offline teaching method is undoubtedly the best choice in the teaching of software operation courses. The purpose of this study is to collect the main influencing factors of students' good or bad learning performance in software operation courses. In order to better serve the construction of online and offline hybrid teaching mode, improve the teaching quality of software operation courses, so that more students can benefit from it, which not only broaden the channels of students to learn software, but also enable students to experience the fun of learning while increasing their knowledge and skills.

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

2. Methodology

2.1 Population and Sample

For the current research, the population comprised about 1100 students of the sophomore, junior and senior students of College of Art and Design of Southwest Forestry University in Kunming city, in Yunnan Province, in China.

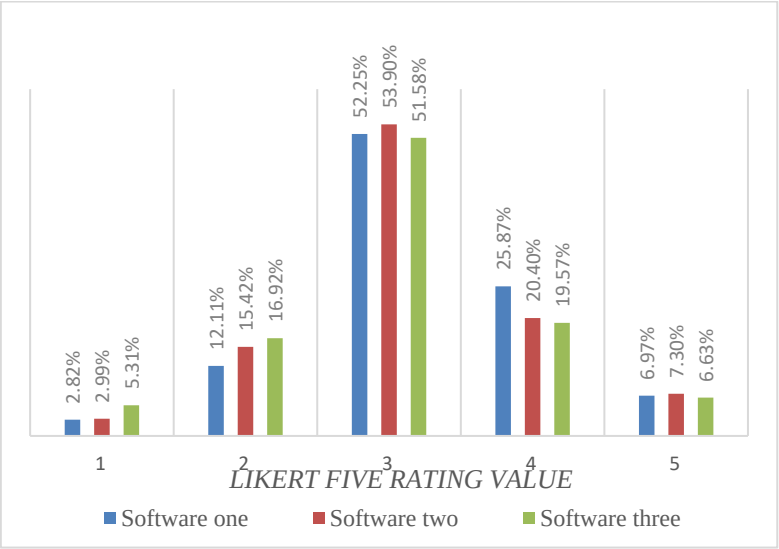

Fig.1 Overall evaluation histograms of the three softwares

A probability sampling procedure will be used to select respondents from the whole population by using random sampling. The questionnaire survey collected data through the website, and a total of 610 pieces of data was collected. Through screening the questionnaire data, two groups of students who mastered software well and those who did not selected. The selection criteria for students with good grades are: 18 students with a proficiency level of 75% or above. The criterion for poor learning is a proficiency level of less than 25 percent, consisting of 18 students. The 36 interviewees were composed of 12 sophomores, juniors and seniors.

2.2 Research Instruments

In this research, the questionnaire was utilized to gather the quantitative data to address the necessary quantitative research questions. The textual data collected by the group interview method were used for qualitative analysis to conclude the main influencing factors of students' learning performance. Mixed research can correct and make up for the deficiency of purely using qualitative research or quantitative research, and summarize and integrate the main influencing factors of students' learning effect of software courses in a more comprehensive and systematic way.

The present study is composed of two parts. The questionnaire and the interview outline are all made by researchers

Part one. The questionnaire is composed of 11 questions in the form of five-level Likert Scale (1= Very Unsatisfied, 2= Unsatisfied, 3= General, 4= Satisfied, 5= Very Satisfied). The main purpose is to understand how well students master the software of their major, as well as the channels, time and motivation of learning software. The questionnaire allows students to list the three major software they have learned in their major and to self-evaluate their mastery of each software.

Part two. Focus group interviews. The researchers inquired about the reasons of the interviewees' software proficiency and the main learning methods, and formulated an interview outline with 10 questions, which was suitable for the interview of 36 students whose software proficiency was above 75% and below 25%. Through the interview, the interviewees' proficiency in software can be deeply grasped, and the methods used by the interviewees to learn software and the problems existing in software teaching can be further understood.

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

2.3 Method of Procedure

There are two stages.

In questionnaire survey experimental stage. First of all, the researchers designed a questionnaire around the core question of "students' mastery of software courses for their major". The questionnaire included 11 questions about students' basic information, the name of the software they learned for their major, their mastery of the software, learning channels, learning time and motivation. Secondly, the questionnaire questions were collected in the form of self-filling online, and the homeroom teachers of each major in the School of Art and Design were responsible for sending the questionnaire links to the corresponding class groups, so as to ensure the validity, objectivity and scientific nature of the questionnaire data collection. Finally, After collecting the data, they were coded into SPSS to process the statistical evidence. The value data of software proficiency is extracted, and the self-evaluation histogram of the total number of students in three grades on software proficiency is drawn, so as to observe the overall level of software proficiency in three grades more intuitively.

In the focus group interview experimental stage. The experiment used semi-structured interview type, and the interview questions mainly centered on the learning experience and feelings of the interviewees in the process of learning the software courses of their major. The interview questions are designed to deeply grasp the methods, existing problems and suggestions of interviewees in the process of software learning from the aspects of students' learning, teachers' teaching and school management. The interview research process is as follows: First, the researcher designs the interview outline according to the research questions; Secondly, the target interviewees are determined on the basis of questionnaire data. Arrange the interview schedule and location; Furthermore, the interview was conducted according to the following interview structure: introduction, warm-up, general questions, in-depth questions, review and summary, concluding remarks and thanks; Finally, the interview data are sorted out, the interview recording files are transferred by voice, and the interview text is analyzed by qualitative analysis software, and the conclusion is summarized.

2.4 Statistical Treatment

The first is the processing and analysis of the questionnaire data. The collected 610 questionnaires were sorted out, 7 invalid ones were removed, and 603 valid ones were finally received. The effective data is imported into SPSS software for processing. In this study, the following data were used to describe the analysis methods: frequency and mean. The data results were visually presented using histograms.

As seen in figure 1, in the 603 valid data, there are about 52% of the students rated their professional software proficiency as average, 29% of the students were satisfied with their own software proficiency (above 75%), and 56% of the students were dissatisfied with their own software proficiency (below 25%). This means that the majority of students in their major of software courses to learn the results of the average. It also shows that the learning quality of the participants in the questionnaire in software operation courses is in urgent need of improvement.

Secondly, qualitative analysis is made on the data of the interview results. Eighteen students with less than 25% software mastery and 18 students with more than 75% software mastery were selected respectively. The analysis results are shown in figure 2 and figure 3. The analysis software used by the researchers is NVivo, which is published by Australian company QSR for qualitative research and analysis. The interview text data was encoded at three levels by NVivo. The results of coding analysis are presented in tabular form.

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

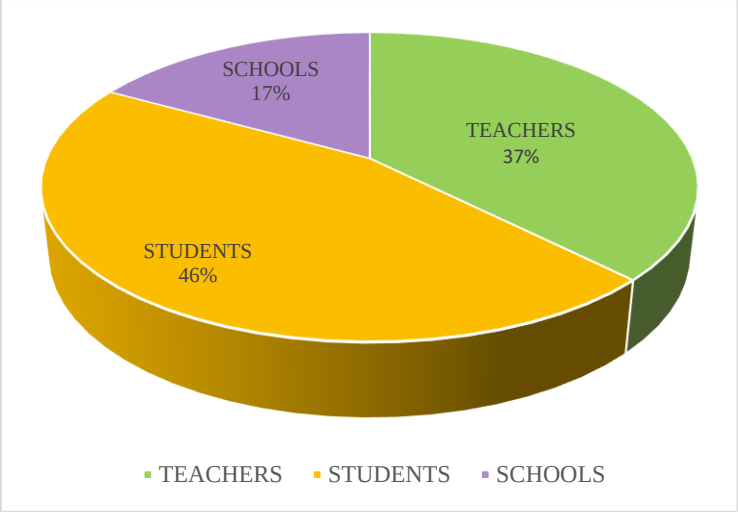

Fig.2 The proportion of influencing factors of students whose software mastery is below 25%

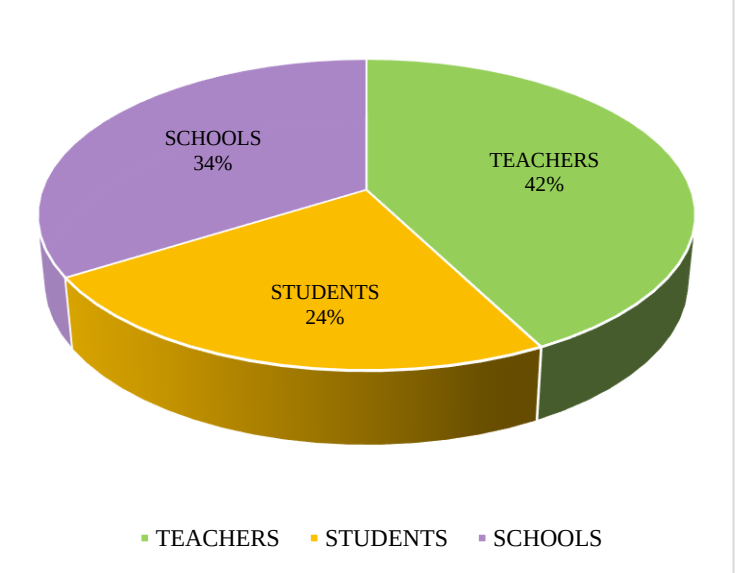

Fig.3 The proportion of influencing factors of students with software mastery above 75%

During the interview, the 36 interviewees were emotionally stable. They could not only tell their methods and feelings about the learning of software courses, but also mention some key problems and relative effective measures. These performances ensure the depth of interview content and the reliability of interview data.

Based on the "localization" principle of Strauss's grounded theory, this research adopts the method of analyzing, summarizing and constructing the theory from the original qualitative data.

Table 1 Interview results of students with software mastery below 25% (Reference node ≥2)

name		file	Reference point
TEACHERS	**Teaching methods**		
	The explanation is too fast to follow	6	6
	It's better to explain it in detail	6	6
	It is better for teachers to record tutorials	5	5
	Network class is good	2	2
	The class failed to remember in time	2	2
	The teacher has his own way of teaching	2	2
	Demonstrate more and coach more	2	2
	Teaching content		
	Speak more and be more careful	3	3
	Learn not good	3	3
	Demonstrate as many examples as possible	2	2

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

	Relationship with Students		
	pay more attention to the poor students	3	3
	Consider the student's actual ability	2	2
	Homework		
	Improve the quality of work	2	2
	Too much homework	2	2
STUDENTS	**Competition Participation**		
	No personal work and no competition	6	6
	Learning methods		
	Sign up for training courses	3	3
	Classroom study + self-study on the website	4	4
	Learning goals		
	Complete the assignment	4	4
	Tests	2	2
	Promoted, then successfully find a job	2	2
	Learning mood		
	Self-regulation and communication with classmates	6	6
	No tiresome mood	3	3
	Own reasons		
	Usually practice less	9	9
	Don't like learning software	4	4
	Practice more and think more by yourself	4	4
	Cannot understand the English of the software	2	2
	Sometimes I can't follow you if I don't understand	2	2
SCHOOLS	**Curriculum**		
	The software course schedule is too tight	8	8
	Not enough time for software courses	6	6
	Teaching conditions		
	There are too many students in class and not enough computers	2	2
	School management		
	More communication and practice should be arranged	3	3

Table 2 Interview results of students with software mastery above 75% (Reference node≥2)

name		**file**	**Reference point**
TEACHERS	**Teaching methods**		
	Too much homework	3	3
	Video recording for students to learn over and over again	3	3
	Teaching too fast in the classroom	3	3
	Teacher guidance is important for students to master software	2	2
	Teaching content		
	The content is too shallow	6	8
	Too much knowledge is taught in the classroom	2	2
	There is too little practice in the classroom	2	2
	The classroom environment		
	The teacher seldom communicates with the students	2	2
STUDENTS	**Learning methods**		
	A lot of practice	3	3
	Learn the basics in class, and study in depth after class	2	2
	Spend more time doing it yourself	2	3
	Learning mood		
	No tiresome mood	3	3
	Own reasons		
	The main reason for studying software is for homework	3	3
SCHOOLS	**Curriculum**		
	The course schedule is so tight that it is impossible to cover all courses	5	5
	There are too few courses and not enough time	4	4
	The course schedule is too long to absorb well	2	2
	Teaching conditions		
	The computer configuration in the school's computer room is too low	3	3
	Lack of opportunities to participate in practice outside school	3	4
	There are too many people in the class to ask questions	2	2

The data in Table 2 are the analysis results of the interview data of 18 students whose questionnaire survey results show that the degree of software mastery is less than 25%. As can be seen from Figure 2, students with software mastery below 25% have poor learning effect and poor learning quality. From the

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

number of nodes, we can see that the main reason why students do not master the software well is that they do not practice after class and use less software. The teacher's operation demonstration is not detailed enough, the explanation speed is too fast, resulting in the students can not keep up with the teaching progress, thus can not fully absorb the knowledge learned.

The data in Table 2 are the analysis results of the interview data of 18 students whose software mastery is above 75% in the questionnaire survey. As can be seen from Figure 2, students with a software proficiency of more than 75% have a considerable self-evaluation and a good learning effect. The learning mode of this kind of students in software courses is based on classroom learning, supplemented by online self-study after class. They tend to reasonably arrange software application and practice activities after class, such as participating in competitions, participating in team practice in the studio and doing private work outside school. In addition, students with good performance in software courses are more anxious about their future jobs, so they will continue to study by themselves after class.

2.5 Conclusions

Based on the findings of the study, the following conclusions were made:

1) Students tend to carry out software operation courses by combining online and offline teaching. The curriculum arrangement is not reasonable, and there is a problem of short time and too tight arrangement, which leads to teachers often fail to take into account the teaching quality of the course in order to catch up with the teaching progress, and students can not learn well.

2) Students can not make full use of all kinds of events to improve the opportunity of software operation; What's more, the classroom learning content that attracts the attention of senior students is divorced from the requirements of the market.

3) The content of basic knowledge taught by teachers is not rich enough, and the teaching content is too simple and not deep enough; There are too many knowledge points instilled in the classroom, which leads to poor absorption and poor learning experience.

4) In the overall teaching, the case demonstration of the teacher is not detailed enough, and the explanation of difficult operations is too fast. As a result, students can't keep up with the teaching progress of the teacher, and the arrangement of homework after class is also unreasonable.

5) The lack of communication and active guidance between teachers and students in software operation courses not only leads to teachers' inability to accurately determine the many questions that students have in the process of software learning, but also leads to students' lack of clear learning planning and goals, and their works are not well reviewed and evaluated and their existing problems are clearly recognized.

6) In the current offline teaching, there are significant differences in the software proficiency of the sophomore, junior and senior groups. Among the interviewees, the degree of software proficiency of the second-year group was significantly lower than that of the third-year and fourth-year group. Generally, the software that students think they have mastered well is the one that has been used or used the most recently. The more frequent the actual operation and software application, the more obvious the learning effect will be.

2.6 Recommendations

Based on the date analysis results and conclusions, the researchers give the following suggestions:

1) School administrators should provide support from funds, equipment and platforms to improve the teaching conditions of software courses. Schools should improve the curriculum system, allocate curriculum time reasonably, and accelerate the implementation of the teaching model combining online and offline.

2) Schools should attach importance to relevant campus competitions to strengthen students' software application ability. Strictly review the training program of students, learning content and the future employment market requirements of unity.

3) Teachers should pay attention to the improvement of their own software operation ability and teaching skills, pay attention to teaching design, learn to flexibly use different teaching methods, so as to make students interested in the class and improve the teaching effect.

4) Teachers should pay attention to the classroom status of students, focus on the demonstration of the

ISSN 2522-6398 Vol. 4, Issue 6: 36-42, DOI: 10.25236/FER.2021.040608

practical operation of the design software, and strengthen students' memory of the operating steps of the software. The assignment in homework after class should pay attention to the pertinence and practicality, in order to enhance the motivation of students to complete the homework.

5) Teachers should strengthen communication with students, pay attention to students' advantages and specialties, point out students' learning directions and help students make learning plans; Students should take the initiative to ask teachers questions, and show teachers the key points and difficulties in learning, which is more conducive to teachers in the teaching with a definite target.

6) Students should actively use the combination of online and offline learning methods, and reasonably arrange time after class to deeply study software. More operations, more exercises and more application practices are the best way to learn software. Lower grade students make full use of classroom and network self-study, while higher grade students consult network resources for in-depth self-study, and improve their ability to use software by participating in competitions and various practical applications.

Acknowledgements

I would like to thank my professor and students for their great help in educational scientific research. The project named Research on Online and Offline Teaching of Software Operation Courses (YB201920) Funded by Educational Research Project of Southwest Forestry University.

References

[1] Flick U. Doing triangulation and mixed methods [M]. Sage, 2018.
[2] Strauss A, Corbin JM. Grounded theory in practice [M]. Sage, 1997.
[3] Jing Chen. Research on the Promotion of Chinese Vocational College Students' Participation in Online and Offline Blended Teaching [J]. International Journal of New Developments in Education, 2020, 2(3).
[4] Wang J. Research on the Design and Practice of Online and Offline Blended Teaching Based on the Concept of Flipped Classroom[C]//4th International Conference on Culture, Education and Economic Development of Modern Society (ICCESE 2020). Atlantis Press, 2020: 813-819.
[5] Ren X, Zhang Z, Geng X. Research and Practice of Online and Offline Blended Teaching Mode[C]//6th Annual International Conference on Social Science and Contemporary Humanity Development (SSCHD 2020). Atlantis Press, 2021: 425-428.
[6] Li Y. Research on Online and Offline Blended Teaching Mode[C]//The 6th International Conference on Arts, Design and Contemporary Education (ICADCE 2020). Atlantis Press, 2021: 297-300.