

Perceptions of Learning Concept Among Malaysian Courseware Developers

Norfadilah Kamaruddin*

Department of Photo Media Creative, Faculty of Art and Design, Universiti Teknologi MARA, Malaysia

Review Article

Received date: 31/12/2015
Accepted date: 01/02/2016
Published date: 11/02/2016

*For Correspondence

Norfadilah Kamaruddin, Department of PhotoMedia Creative, Faculty of Art and Design, Universiti Teknologi MARA, 40450 Shah Alam, Selangor Darul Ehsan, Malaysia; Tel: 03 5544 4003

E-mail: norfadilah@salam.uitm.edu.my

Keywords: Development process, Interface design, Learning concept

ABSTRACT

The success of computer software is based on a series of structured development activities that have been formulated into development process. Towards that, understanding the important of the courseware interfaces as part of the software development process among the developers is also important. Thus, a study was conducted to determine the courseware developers' view in Klang Valley by gathering their understanding on the important of interface design towards development process. The investigation was conducted through two methods: a contextual document analysis and face-to-face interviews with the courseware developers. Findings have revealed that the developers were largely ignores the importance of user involvement and prototyping methods as recommended in the international literature in current development process.

INTRODUCTION

The software development process can be described as a development life cycle. In order to manage and ensure the success of computer software, a series of structured development activities has been developed over many years and these have been formulated into development processes. Throughout the literature, a variety of systems development models can be found and each outlining detailed structures and processes and a variety of activities that are closely linked. While many models are acceptable, depending on the circumstances and the goals of the product, it should be noted at the outset that variant approaches impact on aspects of the outcome^[1-3]. Therefore, an appropriate development model should provide the developer with a systematic, well-disciplined and practical approach to the design, development and maintenance of the software.

Concerning this, interfaces has play an important role in presenting the real info of the computer software. As a tool of communication between the users to interact with a system in delivery particular information, the term of interface design for an interactive multimedia application particularly, is not just simply refer a font size, a button placement or the images that the user sees and feels but including every element of a system such as screen layout (consist of the shape of buttons, the positioning of menus, the display of a warning message, the color applied) and selection modes of interaction. However, most of the courseware developer assuming interface design only a simple thing in the process of courseware development and consequently, quality courseware will not be achieved due to the weak quality of interfaces^[4,5]. This situation can be occurred if courseware developers really have knowledge of the interface design function. This also can actually be addressed if courseware developers refer to the principles and guidelines developed by experts in the field of interface design and expert in the design of multimedia instructional materials. Thus, based on this undstanding, a study was conducted to determine the developers' view in Klang Valley Malaysia by gathering their understanding on the development of interface design towards software development process. The investigation was conducted through two methods: a contextual document analysis and face-to-face interviews with the courseware developers.

INTERFACE DESIGN WITHIN THE DEVELOPMENT PROCESS

To understanding the development process of interface design, there are three classic development models established

in the literature. And, because this research project focuses on interactive courseware for educational purposes, a well-known instructional system development model that is identified in the literature also be reviewed. This includes then: (1) the classic waterfall model, (2) the iterative model, (3) the spiral model and (4) the ADDIE model for instructional system development. Within all of these models, software developers and educators commonly work together to generate the learning content, formulate the learning activities and assessment strategies and determine the feedback on the developed software. The common process applied throughout these four model moreover are included analysis stage, design stage, development and testing stage, and evaluation stage.

No matter which development process is used, the design and production of the interface is particularly important, because users tend to rate the usability of a product on its interface design performance. This is because the interface is their first contact point [6] and users will tend to make judgements on first impressions, which are created by the interface rather than functionality [7]. Indeed, many products are never used because potential users never get past the initial interfaces [8]. Besides, establishing the look and feel of a product (which should balance an expression of the product 'identity' with the sensibilities of the target group of users), there is considerable agreement in the literature that effective interfaces are those that are easy to use and which enable users to achieve their goals in the context of product use [7]. The development team must decide how users will interact with, and navigate through, the software system and choose appropriate methods within the interface to provide instructions to users, and provide multiple types of feedback. Thus, courseware developers need to consider how the interface design affects users' experience, as well as which elements and principles need to be incorporated in a particular context and for a particular audience.

Given this complexity, interface design can be viewed as a complete systems design process in its own right. As [9] and [10] argue, interface design is not something that can take place late in the development process after all the important decisions have been made, but is something that must be continuously addressed throughout the process, usually by a specialist interface designer in the development team. And, to develop an effective interface, a detailed understanding of the users and their needs is required, which should involve as one of stakeholder consultation in the design process itself.

UNDERSTANDING THE CURRENT MALAYSIAN PRACTICES OF INTERFACE DESIGN PRODUCTION IN THE SOFTWARE DEVELOPMENT PROCESS

To understand the current practice of software development process in Malaysia against the international views, 10 courseware developers were interviewed throughout Klang Valley. Interviews moreover have determined that five particular steps are being practiced, which are similar to those most commonly referred to in the literature, which is waterfall model. They include the requirements and planning stage, analysis stage, design stage, development and testing stage, evaluation and feedback stage and finally, the delivery and implementation stage. Moreover, they also initiate a requirements and planning phase up front, and include iterative phases in the development stage and evaluation stage to refine the courseware. This courseware development process is visualised and compared to the waterfall model in **Figure 1**.

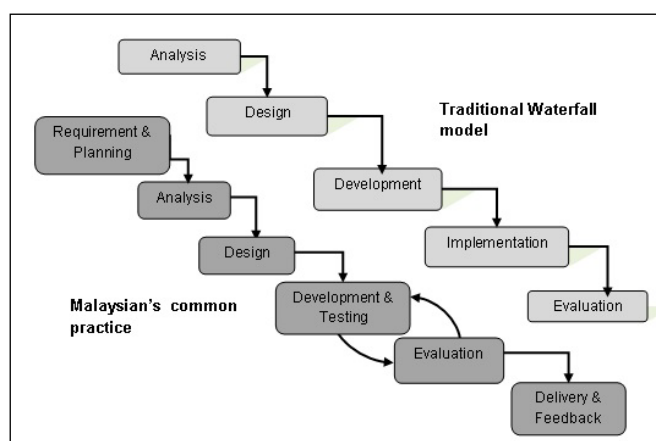


Figure 1. A comparison of steps in the development process in the waterfall model and the development current practices in Malaysia.

However it is important to clarify that the identified differences from the waterfall model (the inclusion of a requirements and planning phase and an iterative phase), does not bring the process in line with the Spiral, Iterative or ADDIE models. Firstly, while the interviews revealed that most of the Malaysian courseware developers (8 out of 10) initiate a requirements and planning phase up front, this includes (1) details of a development work schedule, (2) tasks to be accomplished by the development teams, and (3) the scope of work to be done, and the project manager usually consider factors such as the capability of the team, the budget constraints and completing the task within the given time frame. This detailed and careful planning therefore does not include a user needs analysis, but only an analysis of the ministry guidelines. Secondly, while the process includes iterative phases in the development stage and evaluation stage to refine the courseware, the interviews revealed that the iterative phase is not entirely similar to the iterative process suggested in the Spiral, Iterative or ADDIE models in which, if an error or mismatch is noticed at

any phase of the project by target group feedback, the previous stage is repeated from the beginning. In the current practices of developers in Malaysia, a revision process only occurs if an error is revealed within the development and evaluation stages. The reason provided by the participants is that to make changes then is less time consuming and expensive than the ADDIE model where a change means a total rework. Moreover, the changes are based on internal reviews and requests by the Ministry only. No testing is undertaken with user groups so changes are not based on their feedback. In this regard, it can be concluded that the iterative process being practiced by the developers in Malaysia is simply to ensure that all Ministry requirements are signed off, rather than to produce a product that optimally fulfils end-user needs.

According to the literature, there are three types of analysis that must be conducted before the design process begins: a user needs analysis, a product analysis, and a systems requirement analysis. However, it became clear in the interviews that no participants indicated that they conduct a user needs analysis before starting the design process. Almost all project managers and their development teams simply conduct an analysis of the tender documents and the guidelines provided by the Ministry. As explained by the participants involved, this involves focusing on specific media requirements and instructional specifications in the Ministry's guidelines. Further information is gathered from printed textbooks and other relevant sources in order to achieve the requirements of these specifications. When asked specific questions about why a deeper understanding of the user is not sought, various responses were given. One project manager argued that it is unnecessary, based on the assumption that these needs were understood in advance. He stated that: We are the developers of the interactive educational courseware so we should know the basic requirements of the courseware. So I don't think we need to do a user needs analysis. (Respondent 2)

Other developers cited various pragmatic reasons relating to access to user groups, budget and time. These broad issues are encapsulated by one of the project managers who stated: According to the contract, the period of time to complete the courseware is one year. But then, if you can see, it is not a one year project. In actual conditions we have only 8 to 9 months to finish the project. So we have to speed up ... If you conduct a needs analysis with the user, it will take more time. It is not easy to get permission from the school principal to enter their school. It will involve more additional work and will take you several months to get permission even though they are in the same ministry. Our concern is to complete the process of the courseware development as early as we can, so that we can move on to another project. (Respondent 3)

Another raised schedule constraints in another way, saying that: So far, we haven't yet conducted a user needs analysis with the students. This is because we are familiar with the Ministry project. We also don't have time to concentrate on that because we are not only taking jobs from the government. At the same time we also have to complete other jobs. (Respondent 1)

In relation to the other critical point at which user input is recommended in the literature, namely user testing during and after the design and development phases, the interviews clarified that this does not occur either. The participants reported that the only testing to occur is within their own production team. This testing is performed to ensure that the interface design of interactive courseware has been developed in accordance with the specifications, and that it meets the requirements in the tender documents and guidelines received from the Ministry, as well as perfecting the functionality of the courseware by identifying bugs. As exemplified by one of the project managers: When the lessons are completely designed, in house testing will be conducted by a team leader from each of the different departments. Usually during this process we will make sure that all the functions are working properly. For example, if the voice said "this is a ball" the arrow must be pointed towards the ball not other things. (Respondent 1)

Clearly, the developers try to implement the requests of the Ministry. As one commented: We conduct product testing but only among our team in order to determine the weak points of the courseware, and we modify it before sending it to the Ministry. Generally, this testing is primarily focused on evaluation of the courseware functionality. (Respondent 3)

However this dependency on the ministry guidelines and oversight means that the developers feel that they do not need to take responsibility for any deeper understanding of the users, their needs, or broader design aspects such as the interface design. As one project manager summarised: Actually, the authorization of decisions, especially on interface design, is in the Ministry hands, not ours. (Respondent 4)

This suggests that the developers do not consider it part of their role to undertake user testing, as the Ministry has ultimate responsibility for quality and what defines it. In summary, the courseware developers do not consider end-user involvement necessary or possible within the timeframe, nor their responsibility. However, self-claimed familiarity with the tasks required is not enough to ensure that the interactive courseware is embraced by end-users. As the literature has established, compared with other models of software development, such an approach carries with it a high potential for deficiencies in the content, interface and interaction design of software products.

PERCEPTIONS OF LEARNING CONCEPTS AMONG COURSEWARE DEVELOPERS

As has been established, through reference to the literature, the success of interactive courseware not only depends on courseware developers having an in-depth understanding of the role of interface design (which is crucial to ensuring that users can interact effectively and find what they are looking for efficiently), but also an understanding of the users' needs in terms of the role of the courseware. However, the interviews revealed that most (8 out of 10) developers who participated in this research project lack an in-depth knowledge of the principles of interactive learning.

However, instead of taking the initiative to learn pedagogical frameworks, they rely on the guidelines and requirements of the Ministry, along with general principles for software design and understandings gained from prior involvement in the development of other types of interactive material such as corporate videos and multimedia presentations. This is illustrated by a project manager's response that: We just develop it based on our previous experience. (Respondent 2)

The assumption that this is adequate is revealed in the comment of another developer that: There is not much difference when you are preparing digital learning material. The difference is just a platform of delivery. But at the end of the day, it depends on the user. Either they like to use it or not...it is just about preparing the look of the product. (Respondent 4)

Yet, designing interaction and interfaces for courseware is a very different task to generic software and it is not just a matter of developing interesting graphics and presentation by combining multimedia elements. Interactive courseware has emerged as an instructional technology with the potential to overcome limitations of traditional media and provide the prospect of engaging learning environments with strong visual and interactive elements. Therefore, an understanding of this potential, as well as the understanding that each piece of learning material has different user needs and contexts, is essential.

Reliance on previous experience from different contexts appears to have led the majority of courseware development teams involved in the courseware for the Malaysian Smart School Project (9 out of 10 developers) to assume that the development of educational courseware simply requires an 'improvement' process, which involves converting a printed version of course material into an electronic format. One participant demonstrated his understanding of the difference by this response: Designing interfaces for the computer screen is different from printed design. You cannot simply prepare the interfaces without having some understanding about the overall concept. (Respondent 3)

However, as another, more representative, informant said:

Sometimes we just convert the sample test from the existing textbook into the interactive courseware. It's easier than spending a long time on analysis of the requirements. (Respondent 2)

Designing effective interfaces for interactive material is not only a matter of transmitting information between the computer and individual through graphic presentation and the inclusion of multimedia. Interactive courseware should create experiences for the user that provide them with capabilities, engage them in activities, and create a desire to continue to using the application – all in the interests of ensuring that they understand the educational material presented. By revealing that developers simply convert printed material into a digital format, albeit with multimedia components, the responses gathered in this study suggest that the courseware may fall short in terms of this potential.

DISCUSSION ON IDENTIFIABLE ISSUES WITH THE CURRENT INTERFACE DESIGN PRODUCTION SOFTWARE DEVELOPMENT PROCESS AND THE SKILLS OF DEVELOPERS

Various factors have been put forward by the developers throughout the interviews. The most common reasons for lack of consultation is that they are familiar with the project and the needs of the Ministry, that they are given limited time to complete the scope of the project (that is, the cost and time constraints imposed by the Ministry on the developers preclude a consultation process which would be time-consuming), and that access to end-users is difficult. Moreover, what has also become evident through the interviews is that the courseware developers think that their primary objective is to meet the Ministry's requirements and specifications, and that they do not have the agency to, or responsibility for, understanding the users' requirements. It is understandable that they assume that this is the Ministry's obligation, if the guidelines are presented as comprehensive and binding.

Another phenomenon identified from the interviews is the lack of pedagogical knowledge and understanding of interactive learning concepts among the courseware developers. This is due to the designers in the development team having no experience of, or training in, teaching. However, the courseware developers do not feel an obligation to contribute pedagogical knowledge to the process of interface design production because the content expert in the Ministry team provides instructions and presumably contributes to the guidelines. However this expert does not work with the development team through the process and is geographically and structurally dislocated from the development team.

The contribution of expertise from different fields is particularly important when developing content for educational courseware. System engineers, programmers, graphic designers, writers, and editors must integrate educational material. While curriculum and content might be provided by an educational specialist, also required is expertise in developing effective interfaces for students' and teachers' needs. That is, the development teams in Malaysia not only need a content expert actively involved in the design, but also require an instructional designer who has an understanding of learning models/theories for the effective development of interaction and interface design that engages students with content and facilitates effective testing and feedback. Accordingly, the active involvement of a content expert and an instructional expert to the development team not only helps identification of requirements, but also contributes to more effective performance of the courseware. The dislocation of the content expert and instructional designer from the development teams may be attributed as a potential source of conflict and communication issues between the Ministry and courseware developers, but it is also likely to be a key reason for issues arising in the quality of the courseware.

Given these constraints, the most pragmatic route for developers in Malaysia to take is to simply convert existing printed learning modules into a digital version. It has been well established in the global literature that such a practice affects the effectiveness of the courseware in use. It is therefore a significant problem in the software development system in Malaysia, and can be identified as a factor that strongly contributes to the problems in the interface design, as well as more general issues with the courseware, which impact on its effectivity and uptake. It could perhaps be concluded from this that the courseware developers are more focused on profit considerations than producing an effective product with interface design that accommodates users' needs.

CONCLUSION

Primarily, the development process largely ignores the importance of user involvement and prototyping methods as recommended in the international literature. A number of reasons have been identified. Firstly, the Ministry policy on design and the existing Malaysian guidelines provide no comprehensive explanation about, or requirement to conduct, user need analyses and product testing. Secondly, the limited time and budgets allocated by the Ministry were identified as a sub-theme, and it was argued by the informants that these constraints require the courseware developers to concentrate solely on Ministry requirements rather than contextually specific end-user needs, which are time-consuming to acquire. And, importantly, relationships have not been established or encouraged among the main stakeholders (developers, the Ministry and schools), so meaningful contact between them does not occur.

Out of these primary issues, a number of significant factors arise. Because sole authority for decision-making is held by the Ministry, developers respond to the letter of the guidelines they are provided with, to the exclusion of other types of research or analysis. And, because there is limited pedagogical knowledge within the development team, a misunderstanding among the courseware developers about the concept of interactive learning arises, which may, in turn, lead to the reductive strategy of simply converting the printed learning modules to an electronic format. In addition, because of limited communication between the stakeholders, each function within their tightly designated roles, so sharing and integration of specialist expertise does not occur. For example, courseware developers conduct their product testing within their own team in response to the Ministry requirements, without involving the teachers and students. And a deficit of skills in the development teams leads to the outsourcing of major interface design components, which further fragments the design process and design roles.

These shortfalls in the design and development process of the interactive courseware largely align with the findings of studies in other contexts. As outlined in the literature, there are four main factors that influence the production of interface design. They are the user needs and requirements analysis, functionality, decision-making and creativity. This research study has revealed problems within the courseware development process that may influence all of these factors. Firstly, the current lack of user involvement delimits the user needs analysis and requirements. Secondly, a lack of awareness of interface design and pedagogical theory and strategies among the designers can be related to the effectivity of the interface design and the functionality of the courseware in general. Thirdly, the centralisation of decision-making in the Ministry limits the capacity of developers and teachers to understand each other's need. And finally, the limitation and requirement specificities in the guidelines limit creativity and the potential for innovation.

REFERENCES

1. Rupinder K and Jyotsna S. Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*. 2011; 2.
2. Bostock S. Courseware engineering: an overview of courseware development process. 1996.
3. Lee PM and Sullivan WG. Developing and implementing interactive multimedia in education. *IEEE Transactions on Education*. 1996; 39: 430-435.
4. Norfadilah K. Challenges of malaysian developers in creating good interfaces for interactive courseware. *Turkish online journal of educational technology*. 2010; 9: 37-42.
5. Grabham J. Windows 8 vs windows 7: 8 ways it's different. 2012.
6. Faulkner C. *The essence of human-computer interaction*. England: Prentice Hall, London. 1998.
7. Mayhew D. *Principles and guidelines in software user interface design*. New Jersey: Prentice-Hall. 1992.
8. Galitz WO. *The essential guide to user interface design: an introduction to gui design principles and techniques*. New York: Wiley Computer Publishers. 2002.
9. Dines B. *Role of software engineering in software development: why current requirements engineering is flawed*. Springer, Scotland. 2009.
10. Shneiderman B, et al. *Designing the user interface: strategies for effective human-computer interaction*. UK: Addison-Wesley. 1987.