

E-LEARNING DEVELOPMENT AND ASSESSMENT REPORT: CASE STUDY I-FOLIO

^{1a}HAYDER SAAD, ^{1b}SAIF ULDUN MOSTFA, ²RANA ABDUL KARIM, ³ESRAA SAAD
ABDULMAJED, ⁴RANA JABBAR ABD ALI

^{1a,2}Faculty of Engineering and Built Environment, ^{1b,3,4}Faculty of Information Science and Technology,
National University of Malaysia (UKM), Malaysia

E-mail: saifalsamer@yahoo.com Tel: +601121252493

ABSTRACT

E-learning application is on demand nowadays. It provides learner a place to gain the knowledge in anywhere and anytime. In Universiti Kebangsaan Malaysia, a multi-functional educational tool called I-Folio application has been developed and adapted in their learning and teaching process. Through the application, lecturers and students are connected all the time, regardless where they are. In order to fulfill the requirements of lecturers and students, i-Folio is built under a well-organized process model, Waterfall model. The modules are constructed based on structure principle while user-friendly interface are designed according to user experiences. A bottom-up testing strategy with various test cases has been implemented to verify and validate all the functions in I-Folio application. With this functional educational instrument, learning and teaching process in the university has become more effective and efficient.

Keywords: *E learning, I-Folio, Educational tool, Process model, Waterfall.*

1. Introduction

I-Folio application is an useful educational tool used in Universiti Kebangsaan Malaysia (UKM). It acts as a virtual classroom to provide a space to the students and lecturer in the university. Through this system, lecturer is able to make the announcement to their students, upload the course materials and assignments, design the quizzes and also organize the topic discussion with students. Meanwhile, students can access their course materials at anywhere and anytime, having discussion with their lecturer or classmates regarding the courses, answer the quizzes prepared by their lecturer and get the latest related announcement. Besides that, students are also able to edit their personal information and manage their soft skills details in this i-Folio system. With this system as a part of teaching and learning instrument in UKM, it is more convenient to all the lecturers and students in the class to upload and download their course materials respectively as well as discuss the problems or issues regarding the course easily. In the nutshell, I-Folio is believed to create an attractive and dynamic teaching and learning environment in UKM and it makes the learning process to become more efficiency and effectively.

2. Process Model – Waterfall Model

Process model is a set of more specific processes chosen in developing a software or application. Basically, it is a subset under the big umbrella of software development life cycle (SDLC) which refers to a prescriptive characterization of how software should be developed. The model is used as guidelines or frameworks to organize and structure how software development activities should be performed and in what order. In this context, process model can be represented as a networked sequence of activities, objects, transformations and event that embody strategies for accomplishing software evolution. In general, there are several common process models to fit the SDLC such as Waterfall model, Incremental Development model, Reuse-oriented model, V-model, Evolutionary model, Spiral Development model and others.

In this i-Folio application, the Waterfall model has been chosen. The model consists of five main phases which are Requirements, Design, Code, Test, Integrate and Package. The process flow in Waterfall model is depicted as Figure 1. It is a simple and easy to understand model to the users and developers. Each phase has specific deliverables and a review process without any overlapping. The most significant feature of this chosen model is it able to work well for smaller project especially the user requirements are very well understood. Definitely,

there are some weaknesses of the Waterfall model. One of them is it is very difficult to reverse back and do some changes on your application that was not well-thought out in the concept stage especially the application is in the testing stage. The condition is worsening as there is no working software produced in this model until late during the life cycle. Therefore, the well-understanding of user requirements, well-defined design concepts as well as the appropriate method to assess the deliverable of each phase such as testing at the initial phase are very important when using Waterfall model in software development.

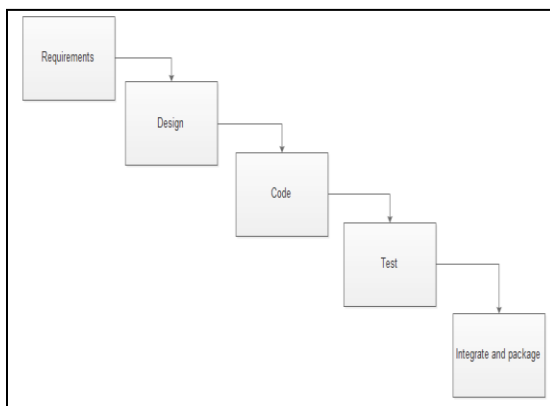


Figure 1: The Waterfall Model

The first phase in Waterfall model is Requirement. This is an initial stage to gather all the requirements from the users before doing some planning on budget estimation and time scheduling. In i-Folio application, lecturer and student are the end-users. In order to understand the needs of lecturer in this application, an interview session is conducted with few lecturers. Meanwhile, a set of questionnaires has been distributed to the students to get their response about their requirements in this i-Folio application. The deliverable of this phase is a complete list of requirements from the end users, lecturer and students in this case.

After gathering the adequate user requirements, design process which is in the next phase in the model can be proceed. The requirements are analyzed and the functions which should be included in the application are identified. Based on the desired capabilities of application, the complete and concise software design and modeling which covers from front-end such as user-interface design to back-end such as database design have to well-constructed.

Once the designs done, user satisfaction is needed in the context. The design especially user interface

design are shown to lecturer and students to get the feedback and comments on the design. This process is iterated until the final agreed designs were obtained. Hence, as outputs of this phase, a complete of system structural model, user interface design and database design were ready to proceed to the next phase in the model, coding phase.

Generally, Coding is more technical phase in Waterfall model. It requires developer to choose a proper programming language such as ASP.net or Hypertext Preprocessor (PHP) to develop the application. Based on the user-agreed design, the application is started to be developed. The application is divided into small units and the coding work is built on unit basic. The testing on each unit is conducted at the end of the development. A software quality checklist is used to ensure the reliability and functionality of the i-Folio application is always in the right path as desired.

Once all the units have been developed and tested, integration and package are the next process in the model. The units are integrated to be complete systems. Before delivering to the lecturer and student to get their feedback on the system, it has been tested again to ensure all the functions in the application are working properly and the interface designs are consistent in every single page.

3. Design Module – User Interface Design

User interface design is the process of crafting a visual language with the focus on the user's experience and interaction. The goal of it is to enable user to accomplish the task with simple and efficient interaction through the application. In the development of i-Folio, the user-friendly interface design should focus on anticipating what lecturers and students might need to do and ensuring that the interface has common elements to be accessed and understood easily as well as to be used in facilitating their actions.

In order to achieve the desired quality of user interface design, the structure principle has been applied in the development. The principle is concerned with overall user interface architecture. First and foremost, the user interfaces should be organized purposefully and presented in meaningful and useful way to lecturer and students. In this context, the related things such as quizzes, info, announcement, groups, tasks, courses materials and discussion have to grouped and placed under the Courses Portfolio section in i-Folio in order to present a more meaningful and well-organized

interface to the lecturers and students. On the other hand, all the elements which are related to the personal information such as the student detail and soft skills were linked and can be accessed thru student home page. By obeying the structure and user interface design principles, i-Folio application is divided into several main sections based on their functionality as illustrated in Figure 2. The function of each section in i-Folio is listed as in Table 1.

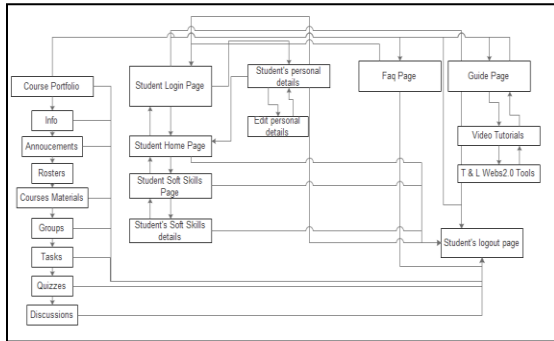


Figure 2: Functional Modules of i-Folio Application

Table 1: Section and its Function in i-Folio Application

SECTION	FUNCTION
Student Login Page	Requires student to input his/her username and password to access the system.
Student Home Page	Displays the important news and reminder to the student.
Student Soft Skills Page	Displays all the soft skills taken by the student.
Student Soft Skills Details Page	Displays every soft skills in details.
Courses Portfolio Page	Displays all the materials and details regarding the courses taken by the student.
FAQ Page	Provide the answer to some common problems met by student in using the system.
Student Logout Page	Allow student exits the system safely.
Guide Page	Provide the video tutorials to the student.

4. Assessment Method – Testing Strategy

Software assessment is a process to evaluate the developed application from all perspectives in order to ensure its capabilities to provide the desired functions are working well. For verification and validation purpose, testing strategy is used as an assessment method to find the errors in i-Folio application prior to delivery to the university. The focus on verification in this context is to ensure every function in i-Folio can be implemented properly while validation is to ensure the application is able to fulfill the requirements of lecturer and students.

In assessing i-Folio development, testing has been conducted based on bottom-up concept which starts from unit testing, integration testing and system testing at the final stage. In unit testing, the focus is to ensure the function in this particular unit is error-free. As an example, for student login function, several test cases are prepared for the testing. The test cases include wrong input of username and password or either, the cases of input (all upper-case, lower-case or mixed), boundary conditions (eg: how many attempts is allowed before account been blocked) and etc.

After testing on every single unit, the related units will be integrated such as student login and logout function. The process of testing on this integration stage is similar as unit testing, but the test cases have been added with all possible scenarios. Example of test cases in integration testing for student login and logout function are whether student is able to revert back to the system by clicking “back” button after logout, is it possible for student to log into the system through different browser or desktop in the same time, and etc.

System testing is a vital process in software development prior to formally deliver the application to the end users. At the final stage, all the main sections will be integrated and linked to each other in order to form a complete i-Folio system. Then, in system testing, developer and users (students and lecturer) are involved. The testing process will be conducted through clicking the hyperlink/button in the application and observing the response of it based on the desired function. A software quality assurance checklist is provided to every lecturer and student for system testing purpose. Any comment about the application particularly in certain function can be written down in the checklist. The responses from

lecturer and students then been analyzed and appropriate actions such as doing some minor changes have been done. Example of software quality assurance checklist is presented in Table 2.

Table 2: Software Quality Assurance Checklist

Testing criteria	Good. Working ?	Failed. Not functioning	Comment
Student Login Function	✓		Student is able to login successfully into the system.
Student Home Page Interface Design		✓	Students might confuse after the login page due to the colour management problem. Too many useless link appear on the top and side of the page.
Student Soft Skill Display	✓		Everything is listed nicely.
Student Soft Skill Details		✓	The page loading is too slow. Students might confuse and do not know what they are supposed to input into the text field. Lack of information or guide line displaying.
FAQ Page		✓	Student is unable to find their related question at there.
Student Personal Details	✓		Student is able to edit all their personal information as intended.
Student Logout Function	✓		Student is able to logout successfully.

Courses Announcement	✓		Lecturer is able to put the latest announcement clearly on this page
Courses Material	✓		The courses material displays nicely and lecturer is able to upload the latest material to the page. The materials are accessible by student without any errors.
Courses Portfolio	✓		Page is able to display all useful information likes amount of members, files, groups, tasks, assignments, quizzes and discussions.
Courses Discussions	✓		Discussions are fully functioning. Discussion among student and lecturer has been created.
Video Tutorials	✓		Video tutorials are able uploaded by lecturer. Student is able to access it.
T & L Webs2.0 Tools	✓		Student is able to access this Webs tools. But the function is too complicated. Student may be not so sure how to use it.
Courses Groups	✓		All information regarding groups in the courses are listed.

5. Summery

In conclusion, i-Folio application has been developed through a well-organized process model. The Waterfall model which consists of six major phases does provide a clear and concise flow in i-Folio development. From the phase to gather the user requirements, analyze their responses, design the

module and interfaces, code the application and at the end to test its functionality, the process is going smoothly and the deliverables from each phase are desired. Hence, i-Folio application, a multi-functional education apparatus is able to bring about optimum benefits to the lecturers and students in UKM particularly to smoother their learning and teaching process.

REFERENCES:

1. Basili, V. R., & Selby, R. W. (1987). Comparing the effectiveness of software testing strategies. *Software Engineering, IEEE Transactions on*, (12), 1278-1296.
2. Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2002). *Fundamentals of software engineering*. Prentice Hall PTR.
3. Kumar, B. S., & Jayasimman, L. (2015). A Comparative Study on User Interface Design of Multimedia Software. *Journal of Advances in Business Management; Vol, 1(1)*, 6-13.
4. Laplante, P. A., & Neill, C. J. (2004). The demise of the waterfall model is imminent. *Queue*, 1(10), 10.
5. Lauesen, S. (2005). *User interface design. A Software Engineering Perspective*, Harlow: Pearson Education.
6. Mayhew, D. J. (1991). *Principles and guidelines in software user interface design*. Prentice-Hall, Inc.
7. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
8. Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. In *Product-focused software process improvement* (pp. 386-400). Springer Berlin Heidelberg.
9. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
10. Singh, D., Thakur, A., & Chaudhary, A. (2015). A Comparative Study between Waterfall and Incremental Software Development Life Cycle Model. *International Journal of Emerging Trends in Science and Technology*, 2(04).
11. Sommerville, I., & Kotonya, G. (1998). *Requirements engineering: processes and techniques*. John Wiley & Sons, Inc.
12. Stefaner, M., Ferré, S., Perugini, S., Koren, J., & Zhang, Y. (2009). User interface design. In *Dynamic Taxonomies and Faceted Search* (pp. 75-112). Springer Berlin Heidelberg.
13. Whittaker, J. A. (2000). What is software testing? And why is it so hard?. *Software, IEEE*, 17(1), 70-79.