# Adaptive Approach for Developing Client-Driven E-Commerce Systems

WANWU GUO

School of Computer and Information Science Edith Cowan University 2 Bradford Street, Mount Lawley, WA 6050 AUSTRALIA

*Abstract* – When clients want a quick deployment of an E-commerce system to maximise their business gain, the traditional systems development models are not fully usable in directing such projects. This is because the client's prioritisation of system requirements may be different from that of the developer's. An adaptive system development model is proposed in this paper for guiding the development of these client-driven E-commerce systems. It is based on the waterfall model but has two concurrent streams in development. Its system-based stream still follows the conventional processes but its component-based stream only iterates over the lower stages. The former delivers the whole system in the end of the project whereas the latter produces prototypes for quick deployments during project. The prototypes can be fully or partly absorbed into the corresponding system groups. The case study of an online trading system development demonstrates that this adaptive approach is effective in guiding the development of client-driven E-commerce systems.

Key-Words: - E-commerce system, Adaptive approach, Client driven, Waterfall model, Prototyping

#### **1** Introduction

In supervising projects on electronic commerce system development carried out by the final-year university students, the waterfall model is commonly recommended to students for guiding their system development. This is because the duration of a student E-commerce project is usually one semester and thus the size of the project is relatively small. This model is especially useful for developing a small E-commerce system whose requirements have been well defined in the commencement and whose deployment date is explicitly set.

In a few cases where clients demand a quick solution to maximise their business benefit, the incremental model is used for guiding such Ecommerce system development. However, experience shows that this model is still not fully capable of dealing with such projects. This is because the client's prioritisation of system requirements may be different from that of the developer's. The former is determined by immediate business benefit whereas the latter is based on the long-term system usability and reliability.

In this paper, an adaptive approach is proposed to handle such client-driven system development for Ecommerce applications. It combines the systembased waterfall model and component-based prototyping together with local increments to deliver the functional components of the system in succession based on the client's prioritisation of system requirements, unlike other approaches that realise the subsystems in sequence.

### 2 E-commerce System Architectures and Development Models

E-commerce system is a kind of distributed systems and usually utilises the client-server architecture to facilitate business activities [1][2][3]. A client is software residing in a user's machine for requesting services through communication network (such as the Internet) from the service provider, or the server. A server is a collection of implementations handling some specific services on behalf of a collection of clients and usually residing in another powerful machine (two-tiered architecture) or other machines (multi-tiered architecture) [3].

No matter whether it is a two-tiered or multitiered system, in terms of functionality, any clientserver system consists of three logical layers: presentation (user interface), processing (business logic), and data (database) (Figure 1). The userinterface level contains all that is necessary to directly interface the user, and thus is implemented in clients. Processing level contains the core functionality for establishing client-server communication, executing business procedure, and interacting with database for data management. Data level contains the actual data and programs that maintain the data [1][3].

These three logical layers can be placed in different physical entities (machines). If only two machines are used, the presentation logic is normally placed in the user's machine whereas the processing and data logics are installed on the server side. In system categorisation, presentation is called as the front-end (sub)system and both processing and data together are named as the back-end (sub)system (Figure 1). When three physical parties host the three logical layers, the data layer is usually placed into an independent machine unknown as database. Although we can still use front-end and back-end systems to describe this three-tiered organisation, it is often categorised as the user interface (sub)system, business processing (sub)system, and database (sub)system, respectively, in system categorisation (Figure 1).



Figure 1 Client-server system architecture and logical layering

Development of such client-server E-commerce system follows the Systems Development Life Cycle (SDLC), which is presented as different system development models. Although variations exist in different models, the five constituents – Analysis, Design, Implementation, Testing, and Deployment – form the backbone of many models [4][5][6].

The simple but yet widely used model for developing small E-commence systems is known as the waterfall model illustrated in Figure 2 [4][5][7]. The five constituents are formed as a chain of sequential stages, with each stage being completed before the commencement of the next stage. This model is useful in developing a small E-commerce system that has well-defined system requirements as the explicit input, and expects a client-server application with predicated behaviours as its output. This system development model does not tolerate any major change in system requirements during development because such a change may lead to the restart of the whole procedure.



Figure 2 The waterfall system development model

The goal of the incremental model is to design and deliver to the client a minimal subset of the whole system that is still a usable system [5][8]. The development will continue to iterate through the whole life cycle with additional increments. This model performs the waterfall in overlapping sections, producing usable functionality earlier. Each subsequent release of the system adds function to the previous release, until all designed functionality has been implemented. This model implies that a high level of inheritance and reusability must be maintained during the development.

Both waterfall and incremental models adopt system or subsystem oriented approaches for system development. The fundamental difference between these two models is that the waterfall delivers the whole system once as the final outcome whereas the incremental allows the whole system to be delivered as multiple installments added to the previous outcome. Each outcome is a result of rerunning the waterfall processes on a subset of the systems requirements. New functions are added to the existing system as constituents of the new subsystem [4][5].

### **3** Adaptive Approach for Systems Development

Experiences in supervising students E-commerce projects indicate that clients are more likely to demand an earlier release of a simple but yet usable system that meets their urgent business needs, rather than a partial system that should support the easy inclusion of the outcomes of subsequent increments. Clients are even prepared to compromise on the higher budget and longer duration for the system development in order to realise their business ambitions. The earlier releases are based more on business-oriented functional components, rather than structural groups.

Another common feature to this kind of Ecommerce system development is that clients still treat the development as one project and want the final system to be deployed by the due date as whole. As long as their business data can be migrated to the new system, clients don't mind whether the new release needs to replace the previous release completely.

To guide the development for such E-commence systems, the author proposes a new system development model shown in Figure 3. Its framework is the waterfall model, but its five processes are split into two loosely linked streams, one following the traditional system approach and the other adopting component-based method. The system stream follows the five stages of the waterfall model for conventional system development. The component-based stream iterates over the Implementation, Testing, and Deployment stages concurrently with the whole system process. This local iteration, or local increment, uses prototyping approach to realise the design of a component group for a quick deployment.



Figure 3 The concurrent system development model (CSDM)

The prototype can be fully or partly absorbed into the corresponding system group, and/or modified to fit into the system design according to the client's feedback from using the prototype. Also the next iteration may include the previous prototype to produce a more functional prototype for the new component group. The intermediate outcomes of the system processes should not be used to improve the performance of a prototype of a component group because the deployed prototype will soon be replaced by the next prototype or the final system.

The key issues to the success of using this new model are to firstly confirm client's prioritization of business requirements, and secondly design the prototypes for the corresponding component groups. The first issue is to confine the potentially frequent changes in business requirements by the client during the quick prototyping because such changes will make the quick prototyping inefficient. Such changes may be accommodated in the system development stream. The design for the prototypes of different component groups should be carried out concurrently with the design for the whole system so that the prototypes can be absorbed into the system as much as possible. This implies that the design for all prototypes must be finalised in the end of the design stage so as to keep the highest possible consistency with the whole system design. This also explains why the prototyping iterates only over the Implementation, Testing, and Deployment stages.

## 4 Case Study: Client-Driven Online System Development for WCE Pizza

WCE Pizza is a recently opened business located near another established pizza shop. Its business operations and offerings are no difference from other pizza shops. Its business promotion strategy is to firstly attract the buyers who are the frequent customers of its next-door competitor by offering lower prices, and secondly retain its customer group by offering them a token-based rewarding scheme. This strategy has been very successful in its gaining of 45% pizza market there.

To make the business sustainable in the long term, WCE Pizza starts a new campaign to increase its customer base. The most important measure is to create a Web-based online system for attracting potential customers living in the radius of five or more suburbs from WCE Pizza. The current paperbased catalogue is effective within two surrounding suburbs. This online system will also make the business operation and management more efficient. A three-tiered client-server system is proposed to be the best option for this system. Customers will use the Web browsers to interact with the system; an Apache Web server containing the Web documents and utilities and business applications along with an email server is hosted in the same computer as the middle processing layer; a MySQL database is used to store all business related data. The interactions among the subsystems will be enabled using PHP and other means. The final system should be delivered as whole in the end of the 22-week project.

However, WEC Pizza cannot wait for five months for the whole system to be operational. A priority list is presented by WEC Pizza based on its business needs. After intensive negotiations, a mutual agreement based on seven instalments is reached between WEC Pizza and the project team. The seven instalments consist of six prototypes and a final system deployment. Tasks and deliveries of the seven instalments are listed in Table 1. Since the prototyping and its intermediate deployment consume extra time and resources, WEC Pizza is willing to extend the project from 22 weeks to 25 weeks and to compensate the additional costs. This client-driven E-commerce system was successfully realised by following the procedure of the adaptive approach shown in Figure 3.

### **5** Conclusion

The adaptive approach is an alternative method for guiding system development, especially for clientdriven business systems. Its waterfall-based framework is easy to follow and understand by the developers. Its component-based stream only iterates over the Implementation, Testing, and Deployment stages, unlike the incremental model that iterates over all five stages. Since the design for the overall system and prototypes is done at the same stage, all the prototypes can be fully or partly absorbed into the corresponding system groups, and/or modified to fit into the system design. This makes the concurrent system development more efficient. The local iteration, or local increment, realises the business separately using prototyping without needs interference with the progression of the overall system development, which ensures the quick deployment of a prototype.

To make sure the success of using this new model, a client's business priority on requirements must be confirmed and the design for the prototypes of different component groups must be carried out concurrently with the design for the whole system so as to keep the highest possible consistency with the whole system.

However, the prototyping and its intermediate deployment require extra time and resources, so this

model will not save the development costs, which must be made aware to the clients in the first instance.

Instalment	Task	Outcome	Delivery
Prototype 1	<ul> <li>enable customers to explore menu and special offers through WEC Pizza Website;</li> <li>provide customers with information for making order by phone.</li> </ul>	Set up a simple Website for WCE Pizza.	Week 4
Prototype 2	<ul> <li>allow customers to register and make order online but to pay goods and/or services face-to-face;</li> <li>keep customer's service records</li> <li>link customers reward points with their purchases</li> </ul>	Set up database server and make it interact with the Web server without online payment facility	Week 10
Prototype 3	<ul> <li>use a bulletin board for promoting special offers and events</li> </ul>	Set up a bulletin board for dynamic messaging.	Week 12
Prototype 4	<ul> <li>alert the registered customers by their emails any new special offers and events</li> <li>inform the registered customers who have accumulated sufficient rewarding point for free offers by their emails</li> </ul>	Set up the email server and customers' mailing list linked with their record in the database.	Week 15
Prototype 5	<ul> <li>link sales with stocks</li> <li>provide the system admin tools for system management</li> </ul>	Add functions to both Web and database servers for tracking stock automatically; Add system management tools.	Week 20
Prototype 6	• make online payment available.	Make sure transactions through Web server, outside banking system, and database system maintain correct states.	Week 22
Final System Deployment	<ul> <li>install and configure the system</li> <li>migrate existing business record to the new system</li> </ul>	Set up the whole system; Make sure the system fully functional.	Week 25

Table 1: Dev	elopment process	es for WCE Pizz	a online business	system using CS	DM

#### References

- [1] J.E. Goldman, P.T. Rawles, J.R. Mariga, *Client/server information systems: a businessoriented approach*, Wiley, 1999.
- [2] D. Ince, *Developing distributed and E-commerce applications*, Addison-Wesley, 2004.
- [3] A. Tanenbaum, M. van Steen, *Distributed systems: principles and paradigms*, Prentice Hall, 2002.
- [4] I. Sommerville, *Software engineering*, Addison-Wesley, 2001.
- [5] R.T. Futrell, D.F. Shafer, L.I. Shafer, *Quality software project management*, Prentice Hall, 2002.
- [6] J. McManus, T. Wood-Harper, *Information* systems project management. Prentice Hall, 2003.
- [7] W.W. Royce, Managing the development of large software systems, *Proceedings WESCON*, IEEE, 1970, pp. 1-9.
- [8] D. Parnas, Designing software for ease of extension and contraction, *IEEE Transactions on Software Engineering*, 5, 1979, pp. 128-138.