April 2006

Course Management System Deployment

Daniel M. Pickett
Worcester Polytechnic Institute

Course Management System Deployment

A Report Submitted to:

Project Advisor: Karen Lemone, WPI Professor

By:

Daniel Pickett

April 27th, 2006

Abstract

The WPI *OfCourse!* course management system provides educators with a set of tools to extend their existent course websites. While this toolset offers great functionality, the installation and configuration of the system is complex and difficult. The project team created a two-tiered installation package that is both simplified and easy to use. User testing was conducted and indicated a significant improvement in the installation procedure. In conclusion, the *OfCourse!* system was improved by the newly created installer.

Executive Summary

As the web evolves and grows, educational professionals continue to pursue the use of online tools. While a variety of software exists, educators of average computer literacy seek a feature-rich application that is easy to install and use. Educators demand a well-designed user interface and a comprehensive set of features.

The *OfCourse!* course management system is a set of online educational tools that can be integrated into existent websites. While the toolset provided is packed with great modules such as calendars, discussion boards, and grade books, the software suite was difficult to install and configure. Although previous attempts were made to simplify the installation process, there was significant variability in terms of how the user proceeded through the installation steps.

In order to create an easy installation procedure, a this project architected a system that provides educators with ease of use without sacrificing functionality. This team set out to provide a deliverable of significant quality in the areas of usability, scalability, and compatibility. With that, the project team also set goals to document the procedure and to eliminate variability throughout the installation steps.

Utilizing best practices of agile software development, the team set out to accomplish their goals. The application was developed iteratively with the help of version control systems. By utilizing these techniques, the project team was able to develop quickly and efficiently.

Upon initial observations, it was found that the application itself needed to be simplified. After a significant portion of the application was refactored, a new

framework was created to abstract database connections with an XML document. Next, a client installer was created that is both easy to use and extensible for future expansion. This client installer handled the writing of the XML document and the transfer of source code necessary to run the system. The server portion of the setup initializes the system with course and authentication information.

In result, the installation process was simplified and became less confusing. While there are some inherent problems with the client installer, the new system provides for a less complex installation procedure with a significant decrease in variability. While user testing indicates a straightforward sequence of steps for installation, problems with the software exist causing the user to become confused. Nevertheless, with the help of documentation and a high speed connection, the installation can be completed with minimal effort.

Table of Contents

Abstract i
Executive Summary ii
Table of Content iv
Table of Figures v
I. Introduction 6
II. Background 8
Educators and Computer Literacy 8
Course Management Software 9
Previous Work 10
III. Analysis & Design 12
Analysis and Initial Observations 12
Objectives 13
Strategies 14
Simplification 16
Database Connection Abstraction and XML 17
The Client Application 17
The Server Side 19
IV. Results 20
The Software 20
Testing 21
V. Conclusions and Recommendations 24
Effectiveness 24
The Future of OfCourse! 25
VI. Works Cited 27
VII. Appendices 29
Appendix A – Directions for Installing OfCourse! 29
Appendix B - OfCourse! Installation Survey 34

Table of Figures

Figure 1. The Basecamp Project Management Application 15
Figure 2. Database Connection XML 17
Figure 3. The Client Side Installer 18
Figure 4. The Server Side Installation 19
Figure 5. An example implementation of a Client Side Configurators 20

I. Introduction

As we move forward with technology and the World Wide Web, teachers are utilizing the internet as a means to communicate with their students. With a growing presence of online educational tools, or courseware, educators are completing the routine aspects of their jobs more efficiently with less aggravation (Moodle Message Boards, 150). While many versions of courseware exist, few programs provide teachers with all of the functionality that they need.

Instructors who wish to use courseware have problems deciding between an out of the box solution versus a customized system. Additionally, computer literacy among educators does not typically include programming experience. For these reasons, a piece of software that is both customizable and easy to install is required to fit their needs. Today, the software is still cumbersome and difficult to configure. For educators all over the world, providing the best of flexibility and ease of use has been a consistent problem since the advent of online course management systems.

Today, there exists a piece of software known as the OfCourse! Course Management System. This piece of software was created by previous MQP teams with the objective of allowing educators to integrate their existing course websites with a toolset that provides students with interactive features such as a discussion board and a grading system. While this program allows an instructor to augment their existing course websites with a set of tools, these tools are not easily configurable and installable..

In this project, the team set out to design an installation package for the existent software that is easy to use and intuitive for an educator of average computer literacy.

While other solutions have been provided, none have fit the needs of an educator in terms of usability and customization. The team endeavored to create a smart piece of software that carefully guides the human installer though a sequence of installation steps.

The installer package requires common practice usability techniques and standards. To make the interface familiar for the user, the project team researched common aspects of installation packages with a particular focus on web application deployment. After the necessary research was conducted, the team built the software with accessibility and fundamental human-computer interaction principles in mind. Additionally, this software provides a framework for future developers to build enhancements and customizations into the installer package. The end result offers an intuitive and flexible method for deploying the OfCourse! Course Management System.

II. Background

Educators and Computer Literacy

Most teachers have a basic knowledge of computers and its workings. The average teacher can check email, create word processing documents, and print classroom documents such as syllabi, course schedules, and other documents that would traditionally be written and photo-copied.

Teachers are in the position of instructing students because they generally have the ability of explaining concepts fairly well to a group of students. Teachers are instructed to instill a feeling of accomplishment and an attitude that anyone can "do anything if one puts his or her mind to it." It is for these reasons that the implementation and installation of the OfCourse online course management should not impose any significant problems with the proper documentation accompanying the program.

The PDF formatted document found on the website of the San Diego County Office of Education explains the different definitions (Introductory, Intermediate, and Proficient) for various computer functionalities. These functionalities include, but are not limited to Word Processing, Database Usage, Presentation Software, and Publishing. This document also clearly points out what can be taught to teachers of each ability level. This can aid in the development process of a help file or document to be included with OfCourse's installation package (San Diego Office of Education). This project will develop an easy to use installation package and will aid in the smooth transition from traditional grading and teaching methods to a more modern approach, even if the teacher has no programming experience.

Course Management Software

A number of complete, open source Course management systems are available on the internet. One is called "Moodle." The website can be found at the URL http://www.moodle.org. This website explains the four different categories of pedagogical principles which help educators create effective online learning communities.

This CMS (Course Management System) seems to be fairly popular overseas and especially in the UK and Germany. These teachers use the CMS for classes of 200+ students. The testimonials seem to be coming from college professors and secondary preparatory schools around the world.

Moodle is difficult to install and configure. In fact, the instructions for installation is seven pages of text! It requires the configuration of Apache web server, MySQL, and PHP to make the program work. Educators typically do not have the skillsets involved to configure such complex applications. Companies like Site Ground (www.siteground.org/moodle-hosting.htm) have based their businesses on installing and hosting courses for professors with minimal computer expertise.

Hosted solutions are becoming popular for web applications that are utilized by end-users of minimal computer proficiency. These applications require zero configuration and typically charge monthly for services offered by a shared web server. Companies like 37Signals (www.37signals.com), Fire Wheel Design (www.blinksale.com), and Blogger.com (www.blogger.com) have experienced major successes offering web applications with minimal setup.

Recently, Clear Function LLC released a hosted solution called Chalksite (www.chalksite.com). Chalksite is "a total web package designed just for teachers, giving you a personal website and tools you actually need without requiring an IT degree to use them" (Chalksite, 1). This application looks like a great way to solve installation and configuration issues for educators.

Previous Work

OfCourse!, a Course Management System (CMS) was developed by two project teams in 2004. It differs from the traditional course management system's approach by having a toolset supplement existant websites (Kelley, Mentz, Saunders, 9). OfCourse allows educators to pick and choose the functionality they wish to use.

OfCourse was developed with MySQL, PHP, and Perl. PHP and Perl are scripting languages that are quite similar in syntax and usage. Both are common in web applications and provide for flexibility throughout the course of development. Two teams in 2004 set out to develop separate components of the CMS in the hopes of unveiling a unified, finished product at the end of their work. The fact that two different scripting languages were used to build the software became very important in the creation of an installer for *OfCourse!*.

Additionally, when designing an installation program, it is important to understand the database system that the program utilizes. The previous project teams chose MySQL as the database system for the *OfCourse!* System. MySQL is a commonly used, open-source, relational database system. It was chosen over a flat file architecture because it "provides protection from race conditions, is significantly easier to program with, and does not clutter up directories with many types of files" (Kelley, Mentz,

Saunders, 28). MySQL is an excellent selection for this project as the database system is cost effective and performs well when paired with web applications. Additionally, PHP and Perl provide easy-to-use libraries for accessing and manipulating MySQL data.

With the language and RDBMS selected, the teams determined and delegated the components of the system. Do, Dube, Le, and Mai developed the following components: Assessment tool, File Exchange Tool, Polling/Voting tool, and Scheduling tool. (Do, Dube, Le, Mai, V) while Kelley, Mentz, and Saunders implemented the Chatroom, Class Listing, Discussion Board, and the Grade Book (Kelley, Mentz, Saunders, 5).

While a setup script was provided, the installation routine left much to be desired. Our team has been charged with the task of creating a more usable setup routine so that educators of average computer literacy can deploy OfCourse onto their website.

III. Analysis & Design

In order to have ensured a proper implementation that fulfills the needs for an *OfCourse!* installation package, appropriate preliminary planning took place. Initially, the project team carefully studied the source code of the system. It was important to have a full understanding of how it worked, particularly as it relates to the setup routines. Once the program was understood, a coherent plan of action was assembled.

Analysis and Initial Observations

After studying the source at great length, it was observed that the setup routines on the server side were much too complicated. It was noted that a much simpler approach could be taken to eliminate errors and server incompatibilities. In the project team's initial assessment, it was also noted that maintaining different versions of the software was becoming increasingly difficult with the onset of many people working on the same source code.

The server side installation routines were too complex to ensure a safe and error-free installation every time an educator would install *OfCourse!*. The code would run a routine to extract an archive, which relied on the server itself to have the proper software and security in place to allow extraction. This lead to initial problems as the shared host, HostUltra, no longer allowed such access to the extraction command. It was noted that this limited the application in terms of its compatibility and scalability.

The program also wrote the database connection information to various files in both the PHP and PERL languages. In order to properly access the database, every file that referenced this connection information had to be modified at install time. This did not allow for the addition of more source code with database interaction without

additional changes to the setup routines, and was therefore poorly designed. It was important for the project team to abstract the database connection into an independent system that did not rely on the writing of specific files with certain programming languages.

The main installation file, serversidesetup.cgi, was much too complicated and added significant variability in terms of how well *OfCourse!* would install. In addition to the extraction and database connection writing discussed above, the setup script checked for upgrades. This upgrade check was fairly complex, again looking at the existence and contents of specific files within the application's folders. This in turn resulted in limitations of reliability and scalability. This check also relied on the existence of certain server configurations which led to compatibility issues and host dependencies.

After the upgrade check was completed, the script then ran a series of SQL queries to build the *OfCourse!* database. There were no validation routines in place to ensure that the database connection was successful. Additionally, there was no feedback provided as to the results of the SQL execution.

The first line of every script has to have a reference to the interpreter that it uses to actually run the code. The installer would update this line on every file in the application. This added another routine that depended on the reference to certain files, and did not provide for maximum scalability and flexibility.

Objectives

After the above problems were observed, a series of objectives were formulated for the installer. These core objectives were created to ensure that each project initiative was in line with the needs of the *OfCourse!* software and the educators that use it.

- **Usability** – the entire installation system should be easy to use for educators of intermediate computer literacy. All parts of the system must provide proper feedback and information as to the progress of the installation.
- **Scalability** – the system should be built to support various web scripting languages. The installer should be flexible to allow the addition of new features and configurable modules. The system should not depend on file layout or the existence of specific files.
- **Compatibility –** the installer should run successfully for a myriad of hosts, whether dedicated or shared.

Strategies

For the purposes of development, agile software techniques were utilized. The agile software movement emphasizes the following:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more. (Manifesto for Agile Software Development, 1)

From these initial thoughts, many techniques were formed. Iterative design was one major adopted aspect of agile software techniques. Each week, the team planned to present a working copy of the application with new features that were prioritized by the stakeholder, Professor Karen Lemone. Iterative design is a "design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a work in progress" (Zimmerman, 1). It allowed the team to stay on track in terms of project scope and the needs of the stakeholders.

Additionally, the project team employed the use of a revision control system known as Subversion. A revision control system is defined as "any practice which tracks and provides controls over changes to source code" (Revision Control – Wikipedia, 1). This system allowed the team to easily collaborate with exterior entities. Additionally, it allowed for the rolling back of changes and easy replication of the source code.

In order to keep in touch with stakeholders and to ensure everyone was aware of progress and plans, project management software was utilized. Basecamp "makes it simple to communicate and collaborate on projects" (Basecamp, 1). The project team utilized Basecamp's To-Do feature to maintain an ongoing, working list of tasks for a successful iteration each week. The messages section also became helpful when broadcasting correspondence to the stakeholders. There is also a file module that allowed us to easily send report documents.

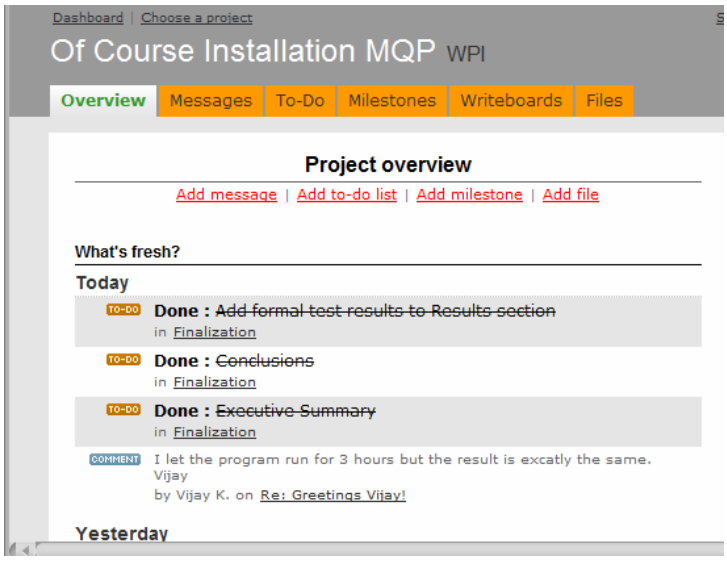


Figure 1. The Basecamp Project Management Application

It was decided that two portions of an installer needed to be created. The client side installer would handle the transfer of source code, and some basic file manipulation. The server side installer would properly set up the database and initialize the system for a

new course. The server side installer had to be simplified and more abstracted so that future enhancements could be added to the system. Likewise, the client side installer had to be flexible to allow new feature sets and configurations.

Later, after deliberating on client side technology, the project team decided to employ the .NET Framework for the implementation. The .NET Framework allowed the team to create a form based application that is easy to use and is widely accepted on Windows based platforms.

Simplification

As a prerequisite to further development, the *OfCourse!* system needed to be simplified. The setup script, setupcgi.cgi, file was simplified to omit the archive extraction, all file manipulation, and security mechanisms. The file was significantly reduced in size. Additionally, there was no need for upgrade mechanisms. Now that the system was placed under version control, there was no need to detect changes in files. Any database changes could now be logged in the single SQL script found in the data folder, as this was simplified with the implementation of version control as well.

To improve simplicity, it was decided that the client installer would handle the security issues such as allowing anonymous execution utilizing the FTP command SITE CHMOD. With HostUltra, this caused significant problems as they would continually disable the FTP command. Later, after the client application was developed, it took the project team a great deal of time to determine what the difference was between the shared host, HostUltra, and the dedicated host used in testing. After diagnosing and researching the issue, it was found that the disablement of SITE CHMOD was atypical among shared hosts. HostUltra was contacted and the issue was remedied.

Database Connection Abstraction and XML

Database connection strings were found in four different locations of the application. These files and functions were added as modules were inserted into the application. In order to contribute to simplification, a universally accessible function and XML mapping was created. It was decided that the client-side installer would handle the creation and writing of the XML file. Having the XML mapping allowed both the PHP and PERL aspects of the program to read the connection strings without being specific to a certain interpreted language. In the future, if a different interpreted language gets used for a new module, all that needs to be created is an interface to read the XML document. This XML document contains the user name, password, database name, and host of the MySQL database utilized for the *OfCourse!* system.

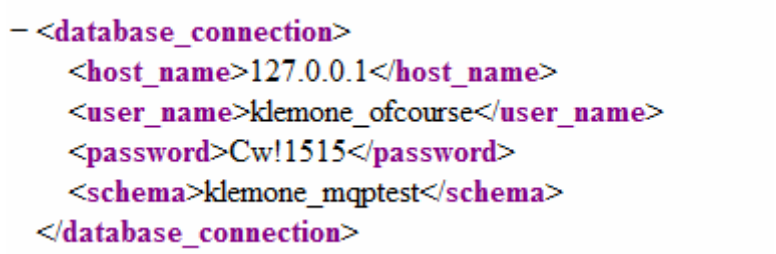


Figure 2. Database Connection XML

The Client Application

The client application was the key aspect of the software. The web application installer was built to handle various configuration mechanisms. What the project team tried to establish was a way for future developers to interface with different database systems and transfer protocols.

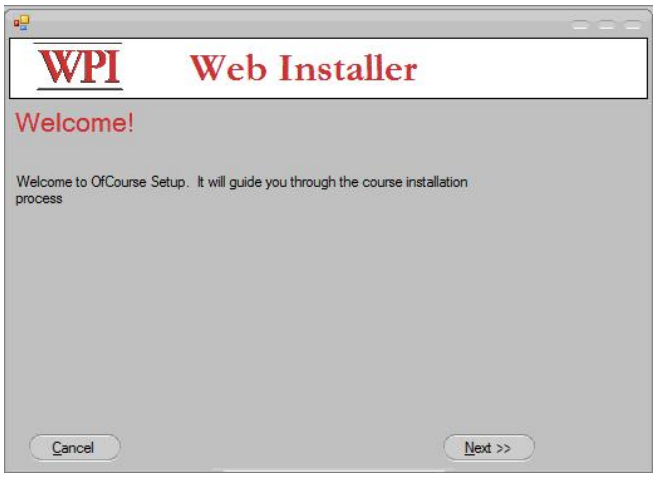


Figure 3. The Client Side Installer

Each panel of the installer configures a different aspect of the software. The first panel asks for the MySQL database information. After the user completes the entry of this information and presses the next button, the program validates the input to verify that the entry was completed. Then, the installer takes the user to the next step. The proceeding panel asks for the web address that you'd like to install into. The next panel asks for FTP connection information. Once that is validated, the user is asked to confirm that he/she is connected to the internet. After the user proceeds, the application begins the installation. It will write the XML file based on the user's input. It will then transfer the source code onto the destination that you specify. A custom FTP client was developed to handle the uploads and security permissions. After the upload is complete, the installer points the user to the specified website to complete the installation at the server side.

The Server Side

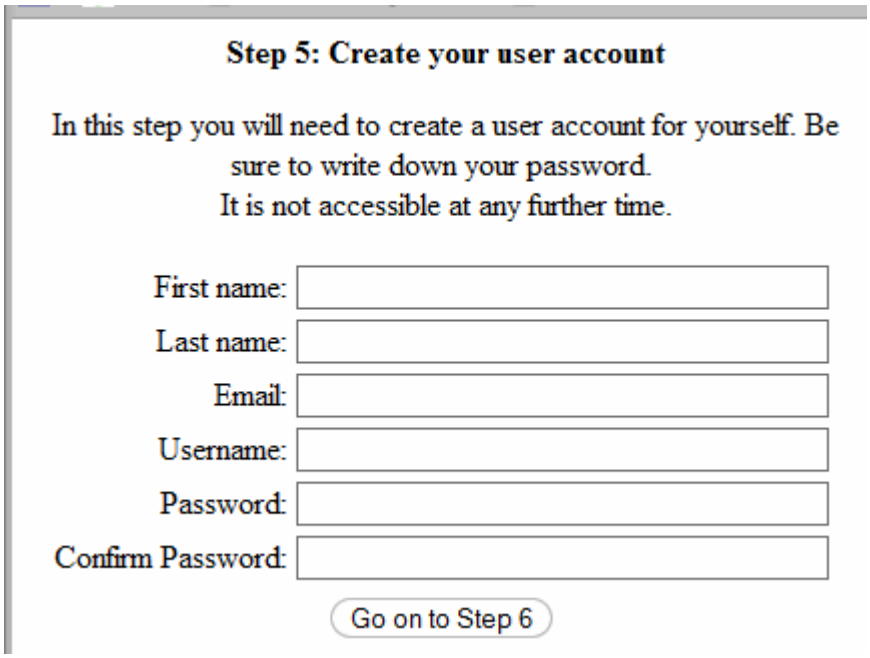


Figure 4. The Server Side Installation

Most of the server side interfaces were left unchanged. The tail end of the prior installation procedure was assessed as straightforward. The project team added some additional elements to the feedback of the programs, however. This provided for better usability. If the user doesn't enter the proper and required information, the installer will now notify them.

The two-tiered installer was designed so that the user could proceed through the setup process without significant guidance and assistance. A short sequential list of instructions was provided (Appendix A) to ensure that the proper chain of events would take place.

IV. Results

The project team was successful in reducing the complexity of the OfCourse! Installation process. Approximately 1,000 lines of code were eliminated from the application, and several steps were taken out of the installation. Providing a client side installer simplified the process of transferring the source code and initializing the database interfaces.

The Software

The client installer, built on the .NET framework provides a system for the installation of dynamic websites. The system was engineered to be flexible and extensible. This means that future developers will have the capability to build new modules for different database engines and transfer protocols. The client installer was found to be straightforward when test subjects followed the directions found in Appendix A. The transfer of source code, however, resulted in the application hanging for the duration of the transaction. In all test cases, this confused the user significantly.

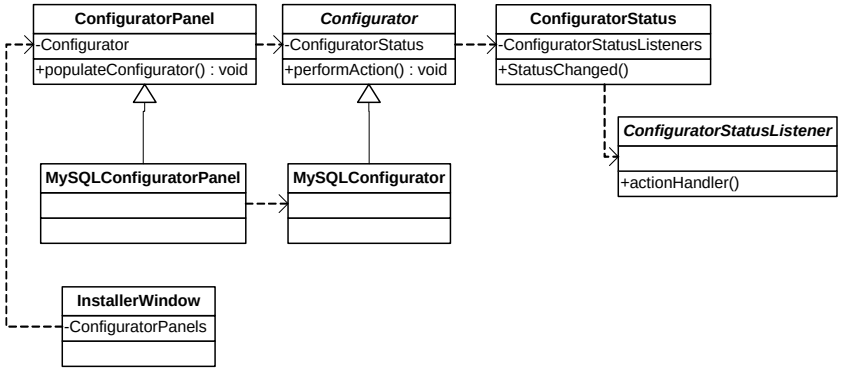


Figure 5. An example implementation of a Client Side Configurator

In the project team's testing of the application, it was found that a repeatable installation process could be attained. This was a significant improvement to the OfCourse! system, as several attempts at the prior installation process would result in different, unanticipated outcomes. Predictable results made the testing and improvement of the application much easier throughout its development. Both a shared host (HostUltra) and a dedicated Linux host were tested extensively to ensure maximal compatibility.

A complete installer for the OfCourse! System was successfully created. With the goals of usability, scalability, and compatibility utmost in mind, the system was designed to be simple and flexible. In order to ensure the accomplishment of these goals, a formalized testing methodology needed to be formulated. It was decided that the project team would initially issue some preliminary, informal testing to improve the application and to refine its supporting documents.

Testing

The project team opted to conduct an informal test with one test subject who will be referred to as Test Subject A. In this initial, informal testing, Test Subject A and the project team began with a copy of the entire client side installer and some basic instructions. The project team attempted to only offer minimal guidance as the subject proceeded through the installation process. It immediately became apparent that a newcomer to the system would have significant difficulty without some supporting documentation. Test Subject A requested that the project team draft directions to guide future newcomers through installation. Because this particular test was conducted with the shared host, HostUltra, guidance was necessary to proceed through the initial

database setup. Once the test subject was ready to open the client installer, it was noted that an identifiable filename would be necessary for the actual executable. After Test Subject A was presented with the proper file, she began completing the client side installation process.

The subject had no problems proceeding to the first area asking for user input, the MySQL database setup. Once there, some of the terminology was unclear for the test subject. It was noted that some of the vocabulary would have to be changed to simplify the experience for the user. Due to the shared host's configuration, it was unclear if the database user and database name fields needed to be prefixed with a user identifier. It was noted that the directions should show sample input as it applied to the shared host.
The user then progressed to the transfer setup form. This was fairly straightforward for the test subject, but the destination directory had to be prefixed with a directory that wasn't known to the subject. This was due to the shared host's configuration. Once completed, the installer began doing its work of writing the XML file and transferring the source code. The test subject was observed to be confused as to whether the application had frozen or not. The project team informed the subject that it was transferring the files and that the behavior was normal. Test Subject A requested that more feedback within the application be offered at this phase of the installation.

The remainder of the installation process was fairly straightforward for the test subject. Due to the subject's familiarity with the server portion of the installation, she was able to navigate through the remainder of the installation quickly. It was concluded that the added feedback and ease of use that a web application provides fulfilled the needs for a usable, server side installation routine.

In further testing, a more formalized approach was utilized including a written survey (Appendix B) and a videotaping of the experiment. Test subjects reported difficulty setting up the databases prior to the client side installation. Additionally, it was observed that Internet Explorer and HostUltra were experiencing some caching errors with certain server-side installation scripts. Despite these difficulties, Test Subject B was able to complete the installation with the help of the documentation. Due to problems with HostUltra, it was recommended by the test subject that a troubleshooting guide be created. This can be found in Appendix C.

Finally, another test subject was given only the instructions and the installer program. Test Subject C was able to proceed through the installer with minimal difficulty. The user reported significant delays and application hang-ups when the transfer of the source code began. The test subject was instructed to leave the installer unattended for a minimum of an hour to allow for the transfer to complete. The user allotted three hours for the transfer to complete. Test Subject C was unsuccessful in installing *OfCourse!*. It was concluded that this was due to a slow internet connection. Thus, this installer requires a broadband connection.

V. Conclusions and Recommendations

After significant observation and study, a formalized installation procedure for the *OfCourse!* System was created. Utilizing agile software techniques and common practice human-computer-interaction procedures for testing, the project team accomplished its goal of establishing a usable, scalable, and compatible solution.

Effectiveness

The installation package should provide for repeatable, understandable, and complete installation routines. Testing indicates that educators of average computer literacy can in fact install the *OfCourse!* toolset easily. Additionally, these tools can be installed on both dedicated and shared hosts. Without the installation tools that were built, creating an *OfCourse!* site was both difficult and unpredictable. Now, educators can continue using *OfCourse!* as they begin teaching new courses. This results in higher educator productivity and satisfaction with the system.

In addition, an added and unanticipated benefit from the project's outcomes was the creation of a framework for the client-side installation of dynamic websites. The client-side software was built in such a way that provides for the addition of more configuration mechanisms. This code base could be further developed to support different transfer protocols such as SFTP and different database engines such as Oracle or Microsoft SQL Server. One could also write code that directly interacts with the server itself making the overall installation even more user friendly.

The Future of OfCourse!

While the installation procedure outlined in Appendix A is significantly easier than what existed before, more work could be done to make the installation even easier. After development and testing, it was apparent that a single interface was desired for ease of use. If there was one application that could set up the database, configure the application for connectivity, and set up the course, that would be ideal. Testing indicates that users got confused going from server to client and then back to server again. An application utilizing web services could potentially do the job, but it would require significant programming that wasn't reasonable for the scope of this project.

Although the client installer provides a robust and simple framework, the FTP library it uses causes significant delays in the transfer of source code. It is recommended that a new library be built to support a hefty load of files. This was one of the most confusing aspects of the installation process.

While the *OfCourse!* system itself provides a wide range of tools, the code is rather bloated and outdated. A significant overhaul of the system's inner workings would provide for easier improvements and functionality enhancements. As both PHP and PERL now have Object Oriented capabilities, it is recommended that future software developers design an approach that is more in line with today's commonplace practices. The system should be re-architected and more thought out in terms of scalability and error handling.

Ideally, *OfCourse!* should be rewritten in one programming language. This would provide for ease of maintenance and the ability to share libraries of code amongst the different modules of the application. Finding a programmer that has extensive

knowledge in both PHP and PERL can be difficult. If the application was written in one language, a specialized programmer could really improve the application from a more abstract perspective. In general, more generalization and stability within the code base is desired.

The shared host, HostUltra, caused significant problems throughout the development of the *OfCourse!* installer. While shared hosts are typically difficult to manage, HostUltra was particularly troublesome because of the unique security practices they had in place. The disablement of the CHMOD command when performing FTP transfers led to significant delays and trouble for the project team. It is recommended that a new shared host be sought out.

VI. Works Cited

Basecamp, http://www.basecamphq.com/

Basecamp is an online project management tool. It emphasizes simplicity over feature bloat, and provides affordable software for small businesses and organizations to manage projects effectively.

Chalksite, http://www.chalksite.com

Chalksite is an online tool for creating course management websites. It is inexpensive and easy to use.

Do, Quan D., Dube, Erin M., Le, Oanh Pham, Mai, Vuong Q. "OfCourse! Transforming Web-Based Courses." 2004.

A previous MQP that created half of the *OfCourse!* Course Management System.

Kelley, Eric Kenneth, Mentz, Brian Craig, Saunders, Eric Joseph. "Development of OfCourse Course Management Systems." 2004.

A previous MQP that created half of the *OfCourse!* Course Management System.

Manifesto for Agile Software Development

This site outlines the guidelines for the Agile Software Movement. Their motto serves as the basis for all agile software initiatives.

Moodle Message Boards, http://moodle.org/course/category.php?id=1

Moodle is an open source course management system. This resource was utilized to gain an understanding of user's needs in a course management system.

MySQL Reference Manual, MySQL AB.
http://dev.mysql.com/doc/mysql/en/installing.html.

MySQL is basically the standard when it comes to large scale website data storage. This means that MySQL has been installed on thousands of web servers and could be extremely valuable in finding out what is involved in setting a web application up like this.

Revision Control, http://en.wikipedia.org/wiki/Revision_control

This WikiPedia article discusses the concepts of revision control and its benefits in software development. It also suggests a myriad of revision control systems that you can implement.

San Diego Office of Education, http://www.sdcoe.k12.ca.us/

The San Diego Office of Education has done some extensive research in how online tools and software can greatly complement the work done in the classroom.

Zimmerman, Eric. Iterative Design, http://www.gmlb.com/articles/iterativedesign.html

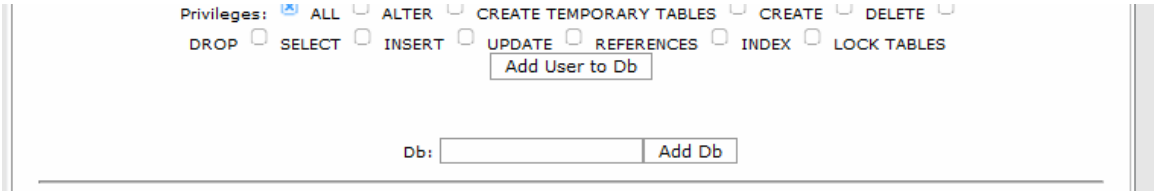
This article provides a close look at the iterative design process and how it benefits software development projects.

VII. Appendices

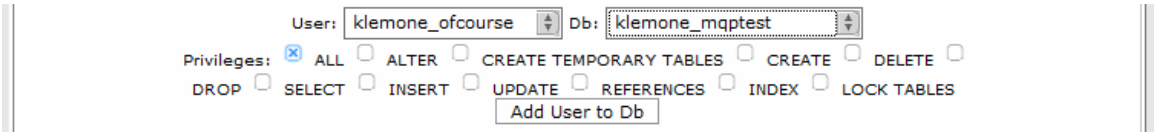
Appendix A – Directions for Installing OfCourse!

1. Before you begin, you have to set up your database.
    a. Navigate to http://www.hostultra.com and click login
    b. Enter your username and password
    c. Click on web hosting control panel
    d. Click on the link entitled “MySQL databases” with the icon shown below.

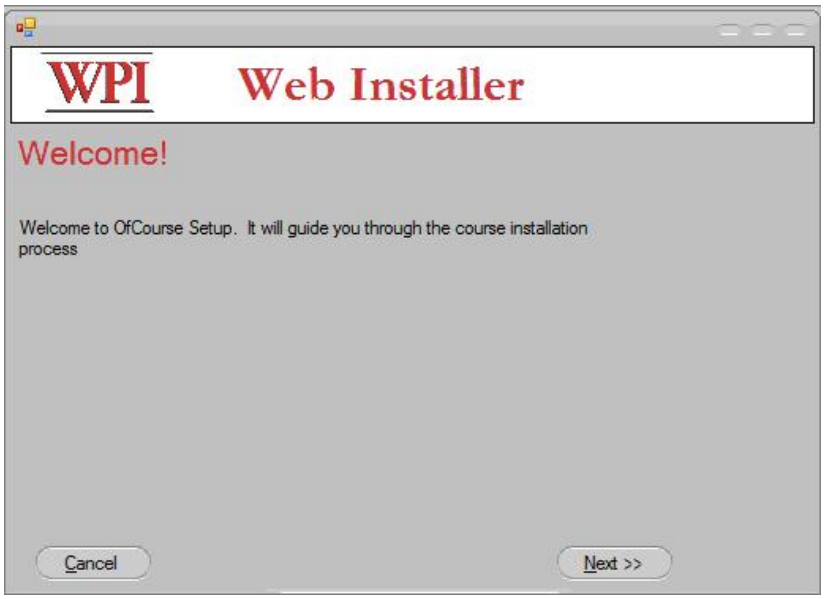
    e. Once on the MySQL databases page scroll down to the textbox that allows you to enter a database name and add it. Write down your database name for future reference.



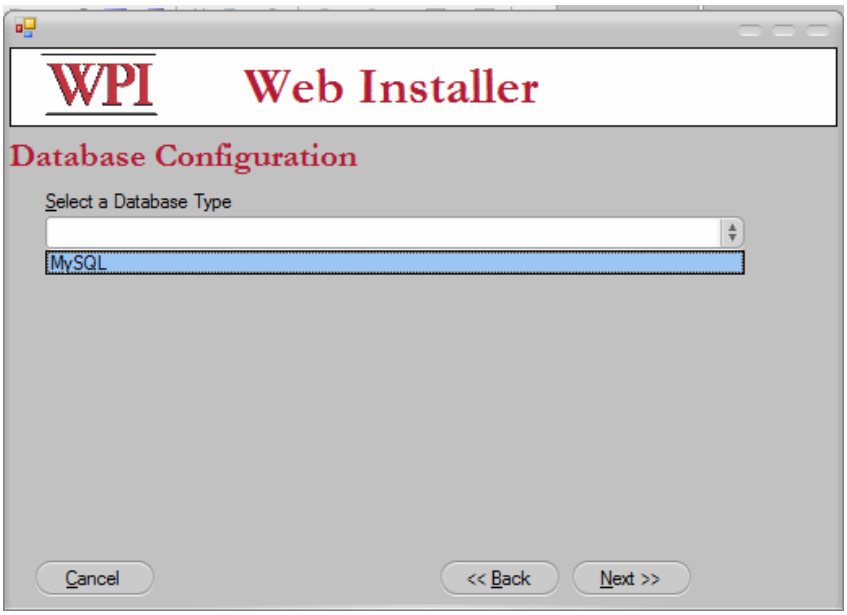
    f. Add a user with all privileges onto the database you just added. Do so by selecting a valid username the database you just added. Write down the user you assigned to access your newly created database. You must also know this user’s password.



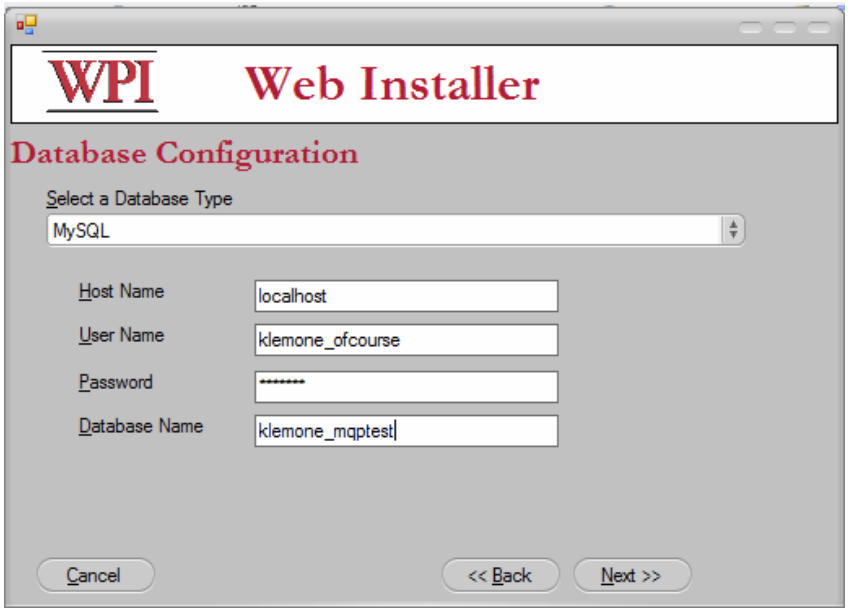
2. Extract the zip into a folder on your hard disk. Open the WPI Web Installer from that extraction location. A welcome screen should appear like below. Click **Next**.



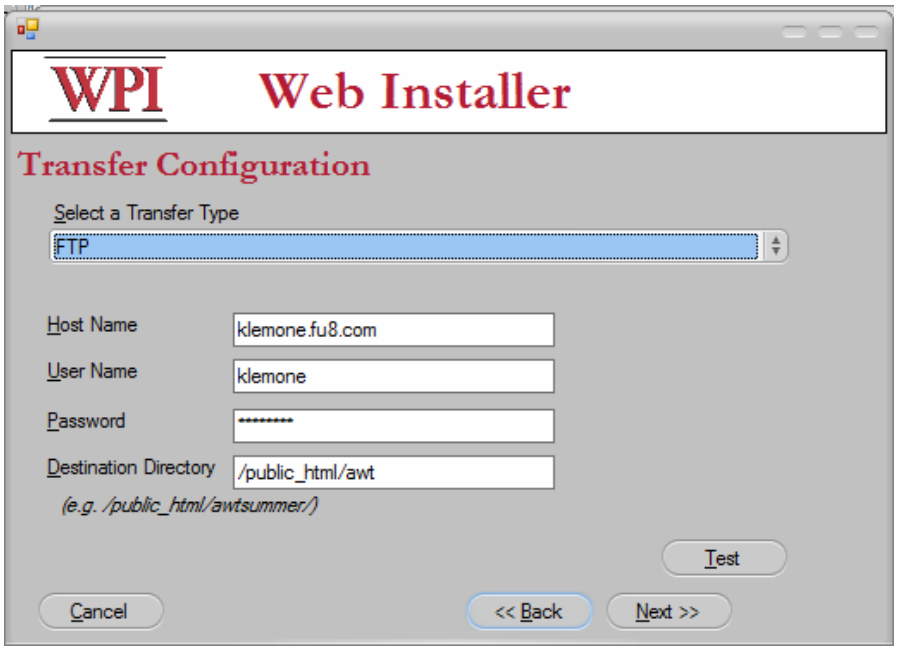
3. On the next screen select a database type. For HostUltra, select MySQL from the select list.



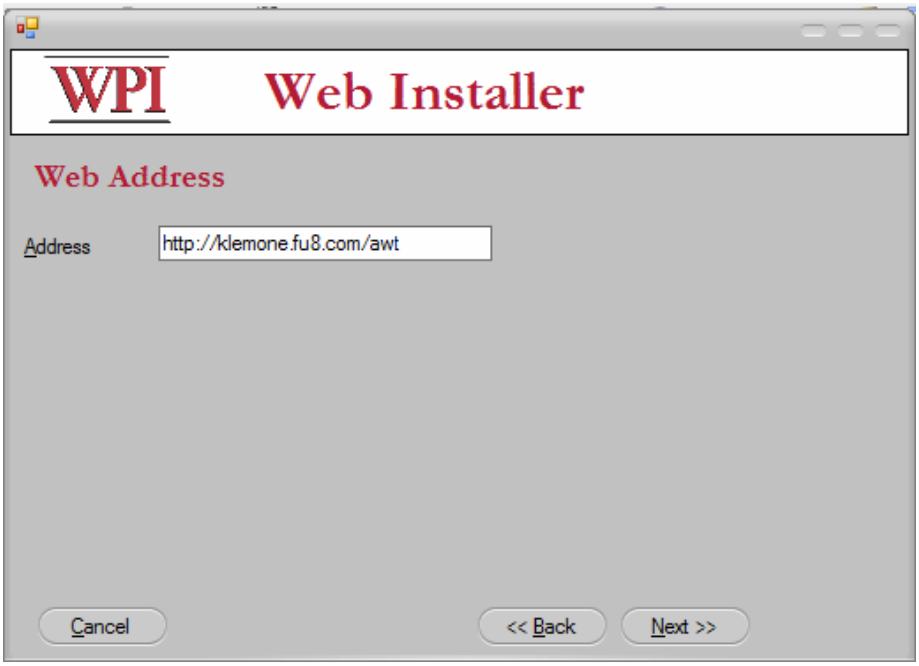
4. Some new textboxes will appear. Enter in the appropriate information that you had written down in Step 1. Click **Next** when finished.



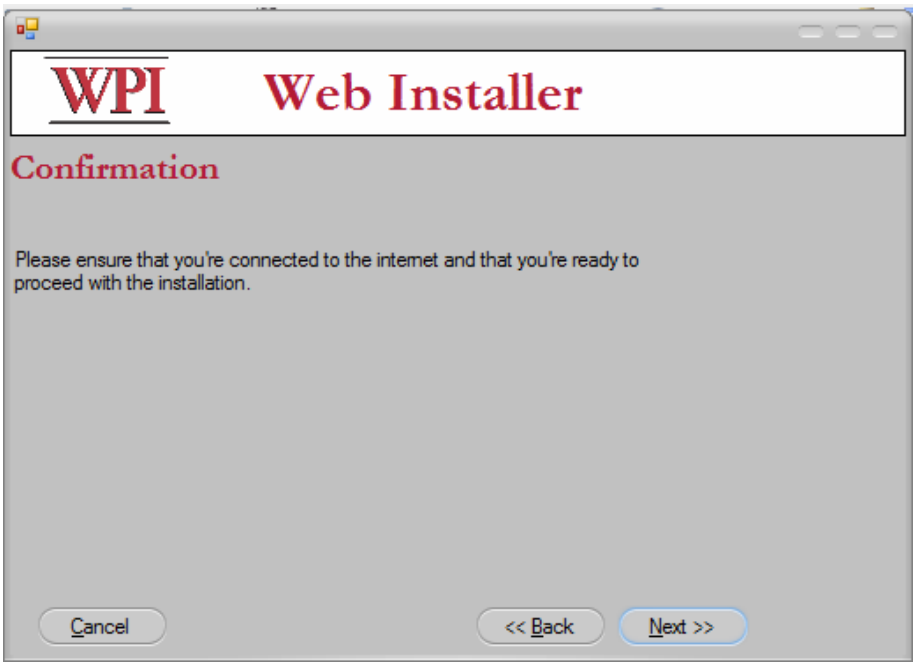
5. Select **FTP** from the Transfer Type select box. Some new textboxes will appear. This is where you provide the FTP information necessary to connect to your site. Also, this determines the location of your tool's site, so remember what you choose for your **destination directory**. Click **Next** when finished.



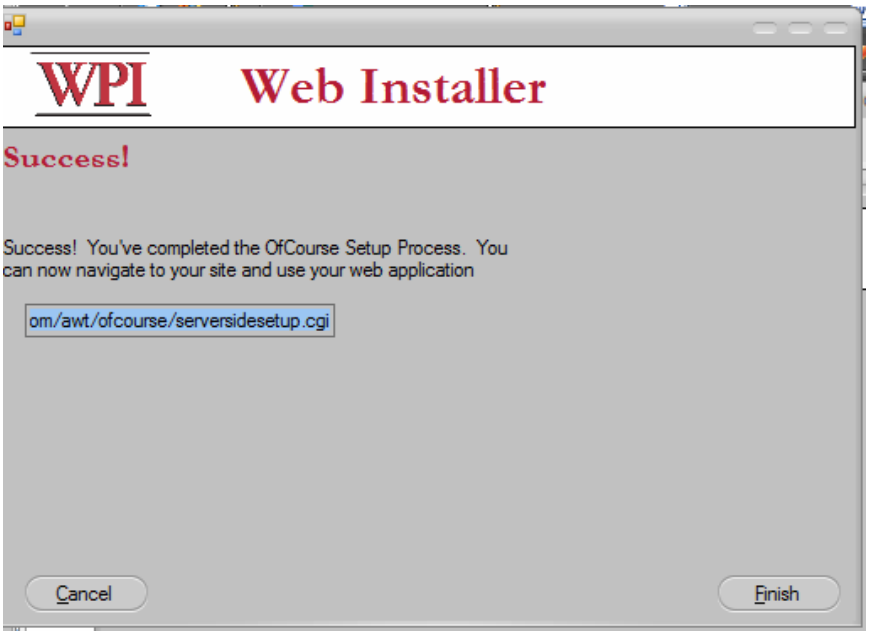
6. Enter the proper address of the tool's site you're going to install. Ensure that this address maps to the FTP location you specified in Step 5. Click **Next** when finished.



7. Please confirm that you have an active internet connection and click **Next**. The installer will then begin its installation routine. Be patient! This takes some time. The application might appear as if it hung up, but it's transferring a large amount of files. This is especially true if you're running the application on Dial-Up.



8. After successful completion, the installer will provide you with a link to complete the installation on the server side. Copy that link to your clipboard and open your web browser. Paste the link in the address bar of your browser to proceed with the creation of a new course website.



Appendix B - OfCourse! Installation Survey

1. Were you able to install a course successfully?

2. On a scale from 1 – 10 (1 = lowest, 10 = highest) how helpful was the instructions document?

3. How did you find the client installer's ability to guide you through the setup process? Please select one option.
    a. Very confusing
    b. Moderately Confusing
    c. Slightly Confusing
    d. Acceptable
    e. Moderately Easy
    f. Very Easy

4. How did you find the server side portion's ability to guide you through the setup process? (The server side portion is the interface you viewed through a web browser) – Please select one option.
    a. Very confusing
    b. Moderately Confusing
    c. Slightly Confusing
    d. Acceptable
    e. Moderately Easy
    f. Very Easy

5. Overall, how would you rate the installer's user interface's ability to guide you through the setup process? Please select one option.
    a. Very confusing
    b. Moderately Confusing
    c. Slightly Confusing
    d. Acceptable
    e. Moderately Easy
    f. Very Easy

6. Please explain any experienced difficulties, if any.

Appendix C - Troubleshooting Guide

Troubleshooting Guide

A. **Internal Server Error**

- Verify that you have correctly set up your database as described in step 1 of the instructions
- Verify that you entered the correct database information in the client installer. This should correlate to the database you created in step 1.
- Check the ofcourse/database/connection.xml document in your web's root directory. Ensure that the proper information is in the XML document.
- Ensure that the permissions on all files in your web root is 755
- Check your server's error log. For HostUltra log in and go to your web hosting control panel. Then click on Error Log to diagnose further problems

B. 404 – Document Not Found

- Most likely the transfer did not successfully complete. Verify that all the files have been properly uploaded.
- Verify that the URL you have specified in your browser corresponds to the path that you uploaded the source into.