Spring 2005

Development and Implementation of an E-Commerce Database Application to Support St. Paul A.M.E. Church

Montez A. Toney
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Acknowledgements

I cannot find words to explain my gratitude to my husband Christopher and my sons Christopher II and Jared. They made many sacrifices in order for me to attain additional learning from Regis University.

In addition to my immediate family, I think I have the best parents in the world. They did not always have a lot of monetary resources but they gave me the foundation to establish a good work ethic, understand that a good education is invaluable and the integrity to work hard and pay for it myself if there are no other resources available.

Finally, I have missed many Sunday worship services due to poor planning on my part regarding the labs during PL/SQL Programming and Advanced PL/SQL Programming. I am dedicating my learning experience from Regis so that I may serve my heavenly father with the same ambition and fortitude I used to complete this goal. I pray that as long as I breathe I can use my education to give back to my church and the heavenly father that has made all things possible for me.

Table of Contents

REVISION/CHANGE HISTORY TRACKING PAGE 5

ACKNOWLEDGEMENTS 6

LIST OF TABLES 10

LIST OF FIGURES / EXHIBITS / ADDENDA 11

ABSTRACT 12

1.0 INTRODUCTION 13

DEFINE THE PROJECT GOALS 14

Items available to sell on the St. Paul A.M.E. Church e-Commerce Site 15

DEVELOP AN INTERFACE EASILY UPDATABLE FOR MODERATE TO NON-TECHNICAL PERSONS 16

PROJECT SCOPE 16

CONSTRAINTS AND LIMITATIONS 17

2.0 REVIEW OF LITERATURE/RESEARCH 19

OVERVIEW 19

Phase I – Systems Analysis Phase 19

Phase II – Design Phase 19

Phase III – Development Phase 20

Phase IV – Testing Phase 20

Phase V – Implementation Phase 20

Phase VI – Training and Maintenance 20

SOFTWARE DEVELOPMENT STRATEGIES 21

Waterfall Model 21

Prototyping Model 24

Spiral Model *26*

Evolutionary Models *27*

Agile Model *29*

Reuse Model *30*

Rapid Application Development Model (RAD) *31*

Development Tools *32*

Microsoft SQL Server 2000 (SQL Server) *33*

MySQL *35*

Visual Studio and Visual Studio .Net *36*

Languages *37*

Commercial Off-The-Shelf Solutions (COTS) *37*

Web Hosting *45*

3.0 METHODOLOGY 47

OVERVIEW OF SELECTED METHODS 47

PHASE I – SYSTEMS ANALYSIS PHASE 47

Software Development Life Cycle *47*

System Requirements *49*

Use Cases *49*

Technical Requirements *58*

PHASE II – DESIGN PHASE 58

System Architecture *59*

PHASE III – DEVELOPMENT PHASE 60

Initial Requirements Modeling and Initial Architectural Modeling *61*

Model Storming *62*

Implementation *62*

PHASE IV – TESTING PHASE 63

PHASE V – IMPLEMENTATION PHASE 65

PHASE VI – TRAINING AND MAINTENANCE PHASE 65

Training *65*

Maintenance *67*

4.0 PROJECT HISTORY 69

WHY AN E-COMMERCE APPLICATION? 69

CONFIRMING AN INTEREST 69

THE PROJECT TEAM 70

PROJECT CHALLENGES 70

WHY THIS WORK TOOK SO LONG 70

REALLY DELIVERING 71

5.0 LESSONS LEARNED AND NEXT EVOLUTION OF THE PROJECT 72

PEOPLE ARE PEOPLE 72

FORWARD THINKING - PEOPLE ARE PEOPLE 72

WOULD THE AUTHOR DO THIS AGAIN? 72

NEXT EVOLUTION FOR THE PROJECT 73

GLOSSARY 74

WORKS CITED 76

List of Tables

TABLE 1 – USE CASE 001 AUTHENTICATE USER ID AND PASSWORD 50

TABLE 2 – USE CASE 002 SEARCH E-COMMERCE DATABASE APPLICATION INVENTORY 52

TABLE 3 – USE CASE 003 UPDATE SHOPPING CART 54

TABLE 4 – USE CASE 004 CONFIRM E-COMMERCE TRANSACTION.....56

TABLE 5 –SECURE SOCKETS LAYER STATUS 66

List of Figures / Exhibits / Addenda

FIGURE 1 – PROJECT TIMELINE 17

FIGURE 2 – WATERFALL MODEL 22

FIGURE 3 – SOFTWARE PROTOTYPING MODEL 25

FIGURE 4 – SOFTWARE SPIRAL MODEL 27

FIGURE 5 – SOFTWARE EVOLUTIONARY MODEL (JIRACHIEFPATTANA) 28

FIGURE 6 – SOFTWARE AGILE MODEL 30

FIGURE 7 – SQL SERVER ENTERPRISE MANAGER 34

FIGURE 8 – MYSQL ADMINISTRATOR 36

FIGURE 9 – COMERSUS BACKOFFICE DEMONSTRATION 39

FIGURE 10 – COMERSUS STOREFRONT DEMONSTRATION 40

FIGURE 11 – CART32 STORE MANAGER DEMONSTRATION 42

FIGURE 12 – ENHANCED SPIRAL MOCKUP 48

FIGURE 13 – UNIFIED MODELING LANGUAGE DESCRIPTION 49

FIGURE 14 – USE CASE 001 AUTHENTICATE USER ID AND PASSWORD 49

FIGURE 15 – USE CASE 002 SEARCH APPLICATION INVENTORY 52

FIGURE 16 – USE CASE 003 UPDATE SHOPPING CART 54

FIGURE 17 – USE CASE 004 CONFIRM E-COMMERCE TRANSACTION 56

FIGURE 18 – PROJECTED CONTEXT DIAGRAM 59

FIGURE 19 – PROJECTED SYSTEM ARCHITECTURE 60

FIGURE 20 – AGILE MODEL DRIVEN DEVELOPMENT (AMDD) 63

Abstract

This project proposal is an endeavor to implement modern technology into long-standing processes at St. Paul African Methodist Episcopal (A.M.E.) Church. The modern technology is e-Commerce. The long-standing process is the method of paying tithes in a collection plate and ordering books, tapes etc. through a designated individual or committee.

The goal of this project is to research realistic solutions and develop a practical plan to implement the project.

Throughout this process, special care is taken to select resources that are moderately priced and straightforward to implement, this is to increase the probability of a successful implementation.

1.0 Introduction

Collecting revenue from tithes and other channels is fundamental to the growth and prosperity of churches. As more payment options become available to the general public, the options evolve into necessities rather than conveniences. An example of this is the evolution of the ATM machine. In the beginning of the ATM era people were afraid to use them, now most people including senior citizens believe they cannot live without them. Thus, more generations are utilizing the web to make purchases and pay bills, largely for the reason that it is convenient.

Although many profit earning companies have employed electronic payment solutions, and while a range of consumers have taken this for granted, there is a significant number of churches and other places of worship that think this convenience is difficult and perhaps expensive to implement and therefore not an option. While this technology may only appeal to segments of a church population that are adept with the Internet and e-Commerce, it is essential for a non-profit entity to provide as many options as possible to collect revenue to keep the organization moving and perhaps attract more members.

To implement this project, resources must be identified to donate hardware and software if appropriate. Also, depending on the needs of the worship center, artwork and other web resources are necessary to esthetically enhance the web site. In addition, the church must designate an individual or team to provide input to the project plan, communicate challenges that the congregation may experience, report problems with hardware and or software and help ensure the project is executed as planned. The church identified to roll out this project is St. Paul A.M.E. Church in Jacksonville, Florida.

Since the membership at St. Paul is large (close to 1,000), it is more likely that the

number of members willing to try this technology would be greater than a church with a smaller population.

Having identified companies that may be willing to provide hardware and networking support, it is likely that this resource can be provided almost free of charge to many churches. Their expenses would include the cost of a monthly Internet Service Provider (ISP), and if elected, a service contract for customized system enhancements.

The requirements defined in this document were derived as a result of speaking with several church members and recognizing opportunities to provide valued added resources in support of the changing resources available with technology. The existing method of collecting tithes and payment for media and events is either cash or check and direct contact with an entity designated to collect payment. As such, collecting tithes and payment for media and events is limited to moments in time where someone is physically available to collect a payment.

In order to accommodate as large a user group as possible, the e-Commerce database application will be developed using a format compatible with Netscape and Internet Explorer.

Define the Project Goals

The most fundamental goal of the project is to provide a resource to sell bibles, other spiritual publications, tapes, CDs and eventually a location that will enable members to pay tithes and make other spiritual donations. As the implementation becomes successful, there may be an opportunity to sell service contracts. Also, depending upon church resources, there may be a need to obtain donated hardware to support the

application.

Items available to sell on the St. Paul A.M.E. Church e-Commerce Site

The items available on the site included bibles and tapes and CDs of weekly sermons and other special events.

Prior to the beginning of the project the Web Ministry had an indexing scheme for the tapes and CDs that fit nicely with normalizing the database to at least third normal form. For example, in addition to the title of a sermon, the label on a tape or CD would indicate a numbering system such as:

- 010103S, or
- 010103A, or
- 010103B

This numbering indicates, January 1st, 2003 and the S, A, or B designates the time of the service. S stands for special as in a service that is not usually scheduled on a weekly basis. A indicates the 7:30 am service and B specifies the 10:45 am service.

A standard reference on the website may include a description such as:

Products in category: Media
Name: How to Handle Life's Valleys.
Description: Tape
Price: $5.00

Develop an Interface Easily Updatable for Moderate to Non-Technical Persons

The interface will be designed using basic Hypertext Markup Language (HTML) 2.0 features and complex features such as frames will be avoided when possible. The site will be formatted using a Cascading Style Sheet (CSS). The main idea of the e-Commerce portion of the St. Paul A.M.E. Church web presence is not to upstage the artwork and general information of the existing site but to provide a place for members to participate in another form of fellowship without having to physically visit the church, mail a letter or make a phone call and with the convenience of conducting these transactions at any time of the day that is suitable for their schedule. The general look and feel of the site will be a conservative color scheme.

Project Scope

The scope of this project covers outlining the development of an e-Commerce application that will easily integrate with the existing infrastructure at the church. The resulting application will also be designed to minimize complex source code and likewise be well commented to assist the web master with updates and changes. In addition to tracking contributions and expenses, the church needs a solution that allows members to purchase various media, including tapes, tickets to events and promotional attire. The church has modest hardware/software infrastructure in place to support the application.

To support the project and the implementation, the church currently has a Technical Assistant to ensure that the requirements and end result meet the needs of the congregation as well as the administration. The project plan and lessons learned documentation from the first roll-out will be used to support development changes and possibly implementation for other churches.

Although the first church targeted with this project does not have an e-Commerce database solution in place, the practical experience of the Technical Assistant leans heavily toward Microsoft products, thus this project will utilize technologies associates with that suite of products.

The methodology of the design of this tool will encompass an interface that will allow non-technical users to add or delete product and upload data to FoxPro. FoxPro is the database currently in use as it is part of the Servant Keeper® application used by the Technical Assistant.

This project will be will be implemented using a multi-phase approach that models the PMBOK (Project Management Body of Knowledge) as outlined in the Effective Project Management guide by Wysocki, 2000.

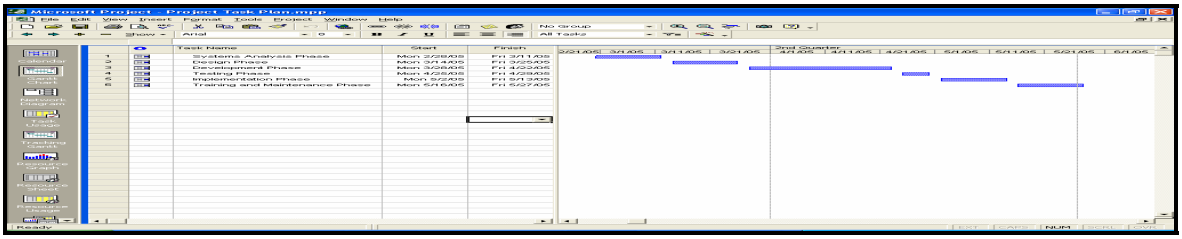


Figure 1 – Project Timeline

Constraints and Limitations

The requirements defined in this document are predicated on the following constraints and limitations:

- The application should be an enhancement to the existing infrastructure and not impede the process of commerce
- Adequate hardware to accommodate MySQL, SQL Server and IIS (Internet Information Systems)

- Availability of reasonable staging environment to conduct testing
- The project will require some data entry on the part of the administrator so that accurate record keeping takes place
- The system will require ongoing feedback to ensure the functionality represented is meaningful and scalable to future enhancements

2.0 Review of Literature/Research

Overview

The project will be conducted using the following phased approach:

- Phase I – Systems Analysis Phase
- Phase II – Design Phase
- Phase III – Development Phase
- Phase IV – Testing Phase
- Phase V – Implementation Phase
- Phase VI – Training and Maintenance

Phase I – Systems Analysis Phase

Documentation necessary to accept payments, and sell product will be collected during this phase. In addition, a vendor that provides Internet security and encryption will be selected. Hardware and software resources will be gathered and prepared for development of the application.

Throughout this phase, the Technical Assistant and Pastor will be provided documentation to ensure that the deliverable is on target with their expectations.

Phase II – Design Phase

Phase II will encompass construction of a Technical Requirements and Design documentation. This includes mockups of the interface, Entity Relationship Diagrams, and detailed descriptions of the architecture of the application.

Phase III – Development Phase

This phase will be accomplished using the Technical specifications as a guide along with the following technologies: SQL Server 2000, ASP (Active Server Pages), VBScript and JavaScript. The design phase will include a regimen of building the application at the end of every development session and documenting results according to the following web browsers: Internet Explorer and Netscape Navigator.

Phase IV – Testing Phase

The testing phase will be accomplished through the development of a formal test plan that will be executed by the Technical Assistant as well as a User Acceptance Test (UAT) facilitated through a selected group of members of the congregation.

Phase V – Implementation Phase

A meeting with a live demonstration will be conducted to provide a question and answer session for the congregation and Pastor. Volunteers among the congregation will be requested to utilize highlighted features of the product to illustrate the ease of use of the interface and likewise resolve issues with conducting e-Commerce transactions.

Phase VI – Training and Maintenance

In this phase, the application will be delivered to the church, self study reference material and reference guides will be provided and the Technical Assistant will receive training regarding managing the database.

Definition of End Product: The end product will be a web enabled e-Commerce database application utilizing available and appropriate hardware resources.

Software Development Strategies

The following software development life cycle (SDLC) methodologies and strategies were researched prior to beginning this project:

- Waterfall Model also known as the Linear Sequential Model
- Prototyping Model
- Spiral Model
- Evolutionary Model
- Agile Model
- Reuse Model
- Rapid Application Development (RAD)

Although there are several SDLC methods in use, some adapted and some fully implemented, it was essential to select something that fit within the scope of the project resources. For example, synchronize-and-stabilize was not an option as there is only one developer on the project, thus no other resource is available to synchronize with.

Waterfall Model

The Waterfall Model is also known as the Linear Sequential Model and the Software Life Cycle. A successful design using this standard encompasses stacking and linking logical events to arrive at the successful delivery of a project. A diagram of the Waterfall Model follows:

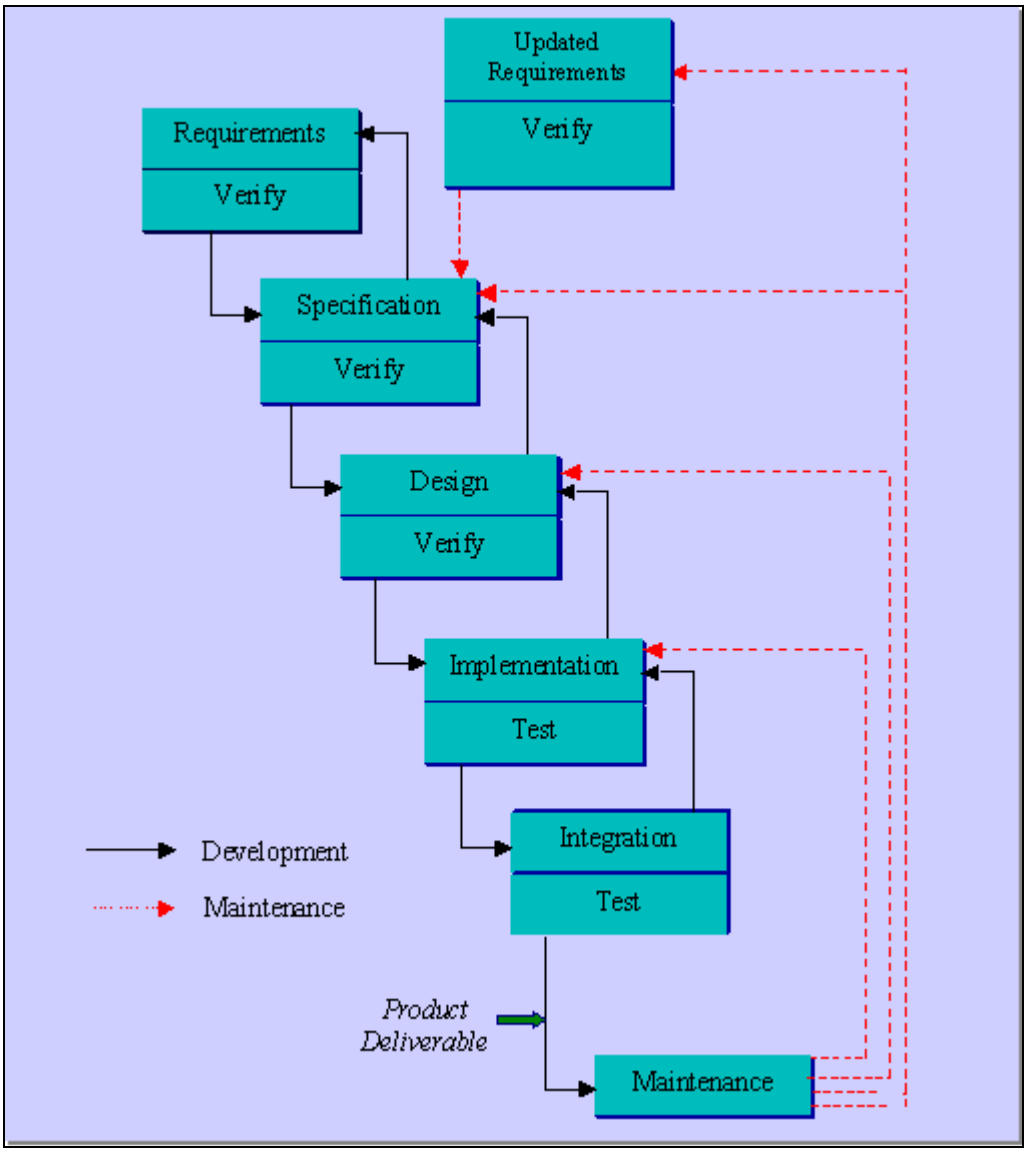


Figure 2 – Waterfall Model

The phases of the Waterfall Model comprise Requirements, Specification, Design, Implementation, Integration and Maintenance. Within these phases the development team following these steps closely should deliver an accurately developed and well tested application. However while these steps are organized in the above specific order, it is this stringency that can cause a software project to go awry.

It is essential when following any development strategy that the Project Manager creates

an environment that fosters flexibility to the changing needs of the project. This includes recognizing when additional factors within the scope of the project must be considered, understanding how to implement features that will satisfy the client and knowing when changes to the project and development strategy will position the deliverable in a favorable situation. Common sense in the matter of delivering Information Technology (IT) Projects can make the difference between just following a methodology to the letter and delivering a quality product. It is imperative to recognize that the deliverable is the technology solution rather than the software methodology.

Below is a passage taken from the University of West Indies at Cave Hill site outlining details of the advantages and disadvantages of the Waterfall Model:

Advantages

- Testing is inherent to every phase of the waterfall model
- It is an enforced disciplined approach
- It is documentation driven, that is, documentation is produced at every stage

Disadvantages

The waterfall model is the oldest and the most widely used paradigm. However, many projects rarely follow its sequential flow. This is due to the inherent problems associated with its rigid format. Namely:

- It only incorporates iteration indirectly, thus changes may cause considerable confusion as the project progresses.
- As the client usually only has a vague idea of exactly what is required from the software product, this WM has difficulty accommodating the natural uncertainty that exists at the beginning of the project.

- The customer only sees a working version of the product after it has been coded. This may result in disaster any undetected problems are precipitated to this stage (The Waterfall Model).

Prototyping Model

Prototyping is a methodology that allows the participants in the project to present a working model of the final deliverable prior to launching a full-scale project. In order to successfully apply this approach, the project team must have excellent communication skills not only among each other but also with the customer expecting the deliverable. There must also be some documentation available that clearly outlines the expectations of the software. Thus it is essential that key elements are identified prior to turning over prototype instructions to a development team. Following is a diagram of the Prototyping Model.

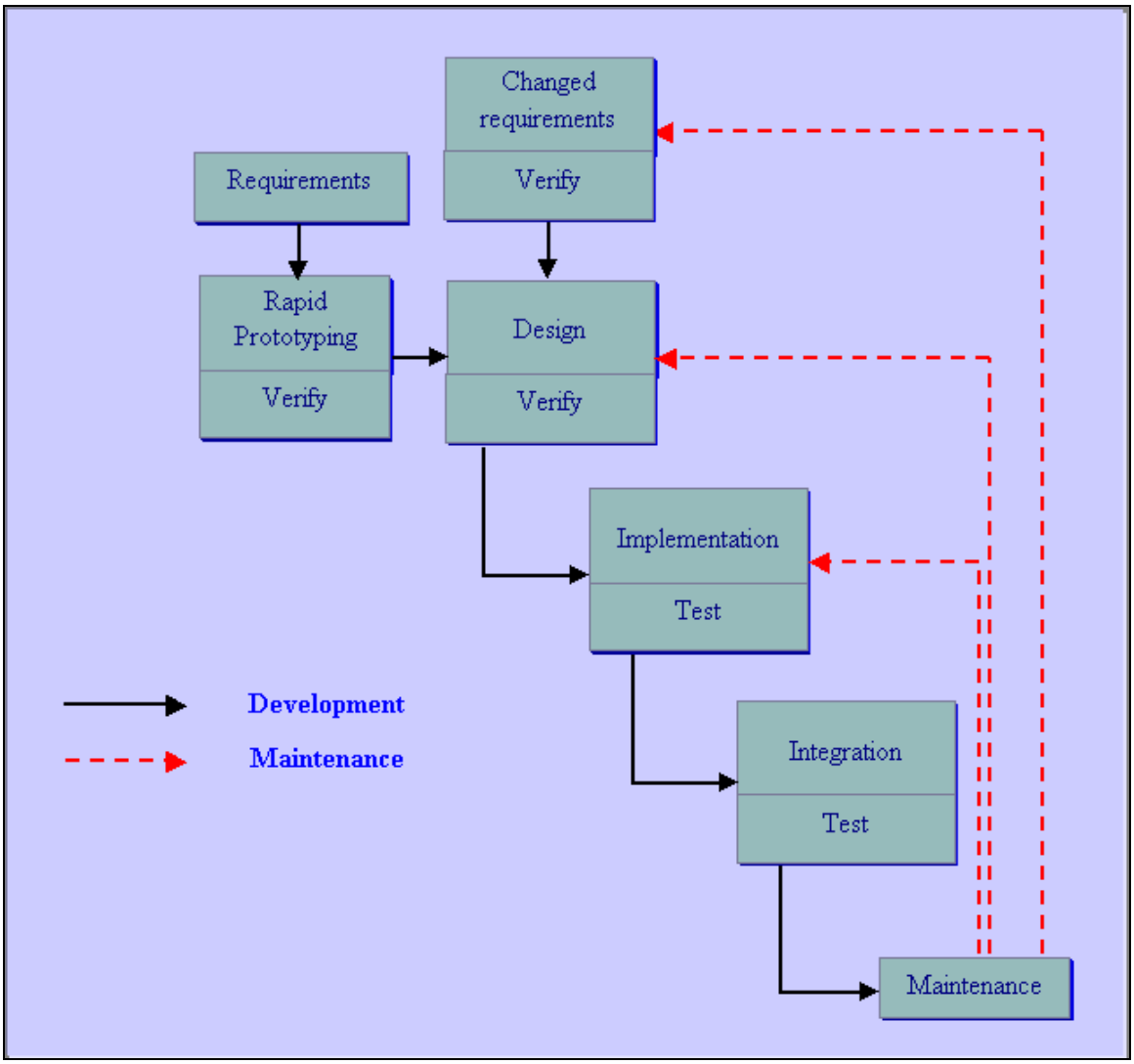


Figure 3 – Software Prototyping Model

As described by the University of West Indies at Cave Hill, the disadvantages of prototyping include:

1. Often clients expect that a few minor changes to the prototype will more than suffice their needs. They fail to realize that no consideration was given to the overall quality of the software in the rush to develop the prototype.

2. The developers may lose focus on the real purpose of the prototype

and compromise the quality of the product. For example, they may employ some of the inefficient algorithms or inappropriate programming languages used in developing the prototype. This *is* mainly due to laziness and an over reliance on familiarity with seemingly easier methods.

3. A prototype will hardly be acceptable in court in the event that the client does not agree that the developer has discharged his/her obligations. For this reason using the prototype as the software specification is normally reserved for software development within an organization.

To avoid the above problems the developer and the client should both establish a protocol, which indicates the deliverables to the client as well as contractual obligations.

In both versions the prototype is discarded early in the life cycle. However, one way of ensuring that the product is properly designed and implemented is to implement the prototype in a different programming language from that of the product (Adrian Als & Charles Greenidge).

Spiral Model

According to Whatis.com Target Search, one of the most widely used models for large and expensive projects is the Spiral Model. The Spiral Model was adapted from the "prototyping and waterfall models" (Words-to-Go: Software Development). This methodology incorporates various stages and developmental milestones and as described by James R. Chapman, "the spiral methodology reflects the relationship of tasks with rapid prototyping, increased parallelism, and concurrency in design and build activities.

The spiral method should still be planned methodically, with tasks and deliverables identified for each step in the spiral" and is depicted as follows:

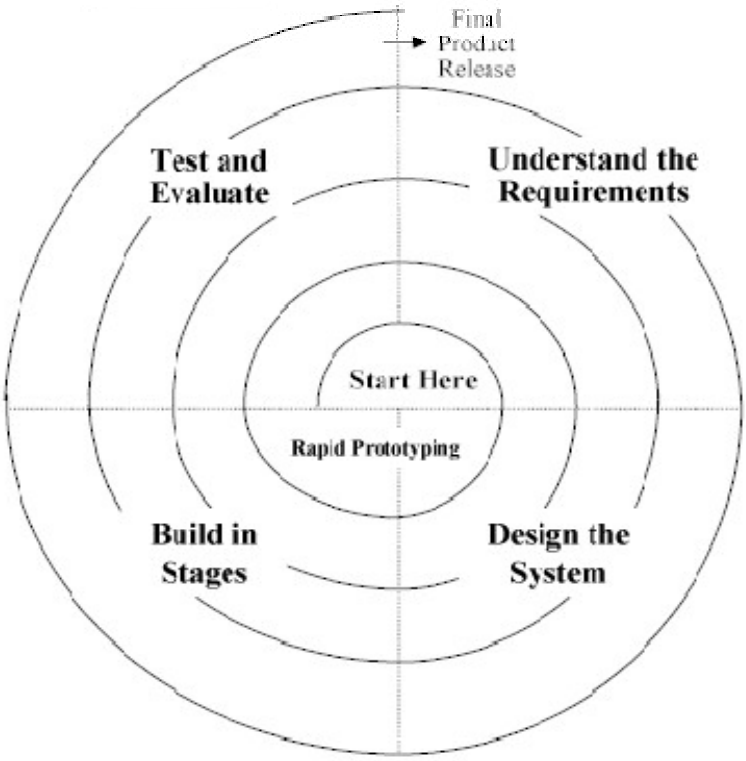


Figure 4 – Software Spiral Model

(Software Development Methodology)

Evolutionary Models

The Evolutionary Model is best applied when a project team determines that the final deliverable is not a stable product but one that matures as the project develops. Consequently sometimes software applications are not completely thought out but left to grow as the project team and customer discover together what the result should be.

Thus the Evolutionary Model is the implementation of a "high-level" idea that is

constantly narrowed down into a polished finished product.

The International Journal of Computer and Engineering Management details a project conducted in Thailand and is focused on the Thai culture staying competitive as other "developing" countries bid for business and develop systems for companies seeking outsource solutions.

While this model has been one of the least likely choices for this project the author Jirachiefpattana, explains the benefit to his project and implies how this model works well with his culture. "An evolutionary development approach should be used. *Our project* is likely to be more successful if an evolutionary development approach is used. In such a volatile *culture* traditional operational development approaches are not relevant. Further, they are antithetical to system success" (Jirachiefpattana).

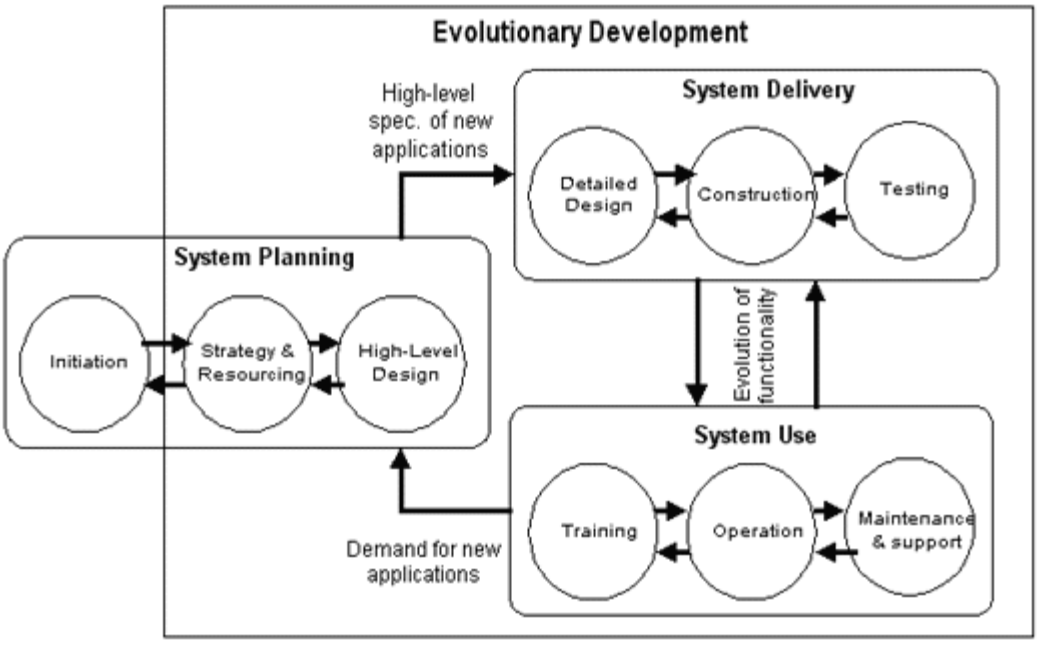


Figure 5 – Software Evolutionary Model

(Jirachiefpattana)

Agile Model

The Agile Model as described by Whatis.com “calls for keeping code simple, testing often, and delivering small, functional bits of the application as soon as they're ready. The focus is to build a succession of parts, rather than delivering one large application at the end of the project” (Words-to-Go: Software Development).

This model would be beneficial in a situation where the final deliverable is a large application that could take several months to fully develop and the customer really needs some type of relief in the interim.

This approach is a competent solution and a likely choice for this project as during the progression of the project, feedback from church members will influence how quickly components of the project are delivered.

The following depiction is from a source that specializes in the Agile methodology. This mock-up is enhanced to reflect the way they have adapted the method to meet their needs however notwithstanding it is a very good demonstration of the process.

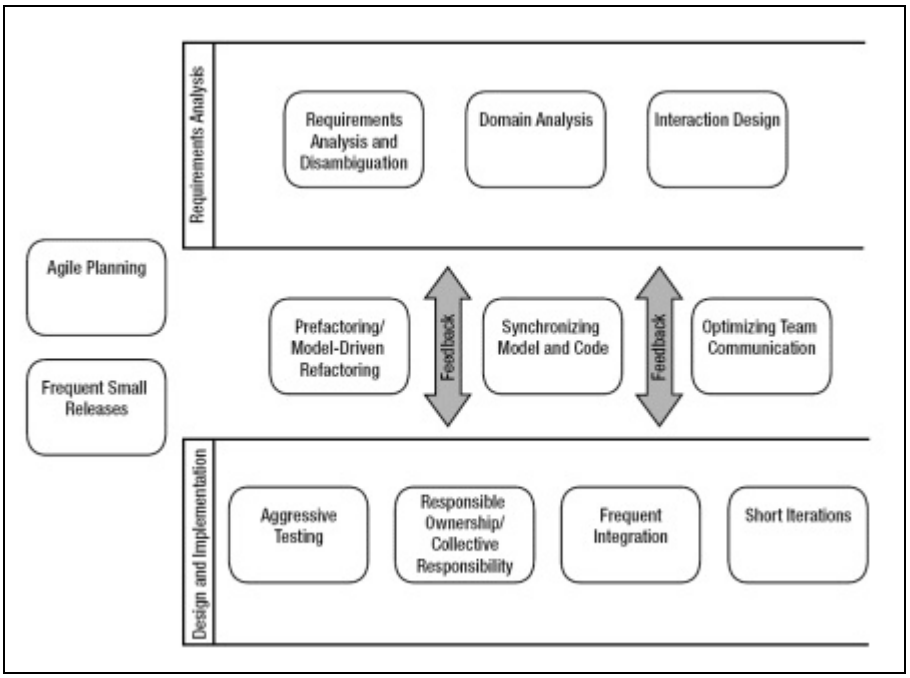


Figure 6 – Software Agile Model

(Agile ICONIX Process)

Reuse Model

Reuse is the circumstance where components of a project are reused to implement another project. Reuse is open not only to software objects and components but also requirements and design. In addition to expediting development time, reusing software elements can be a significant cost savings to the project.

There is a caution to reusing software as described by David B. Stewert, and is labeled #25 as one of the top 30 pitfalls for software developers:

> Code that is not designed for reuse will not be in the form of an abstract data type or object. The code may have interdependencies with other code, such that if all of it is taken, there is more code than needed. If only part is taken, it must be thoroughly dissected, which increases the risk of

> unknowingly cutting out something that is needed, or unexpectedly changing the functionality. If code isn't designed for reuse, it's better to analyze what the existing code does, then redesign and re-implement the code as well-structured reusable software components. From there on, the code can be reused. Rewriting this module will take less time than the development and debugging time needed to reuse the original code.
>
> A common misconception is that because software is defined in separate modules, it is naturally reusable. This is a separate mistake on its own, related to creating software with too many dependencies (30 Pitfalls For Real-Time Software Developers, Part 1).

Reuse may be an option for this application as the nature of the deliverable is a data-driven e-Commerce solution. There are many commercial off the shelf (COTS) products that are both reliable and trusted. Considering the actuality that the audience receiving this application is skeptical about the concept of e-Commerce, it would be sensible to seek a solution with a trusted reputation.

Rapid Application Development Model (RAD)

RAD is a method adapted to maximize the ability to deliver component based applications utilizing a "short development cycle." According to stylusinc RAD incorporates the following phases:

- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation

- Testing and Turnover
- Component Assembly Model

In addition to emphasizing “speed” this process assumes many variables are in place such as the utilizing of a “RAD tools such as VB, C++, Delphi” etc. Thus RAD does not skip essential steps rather it is a high level implementation available when much groundwork is in existence, for example tested components and developers experienced with object oriented (OO) languages (Software Development Life Cycle (SDLC)).

Development Tools

Although the idea and implementation of e-Commerce is new to the church, the proposal of a database is not. Keeping in mind the existing resources available within the Servant Keeper® application and the financial limitations of the church the following aspects were considered:

- Compatibility With FoxPro (the database that ships with Servant Keeper®)
- Facility to Copy Existing Membership Database
- Inexpensive and Ease of Operation

With the aforementioned specifications the following databases and development tools were analyzed:

Databases:

- Microsoft SQL Server 2000
- MySQL 4.1.11

Development Tools:

- Visual Studio

- Visual Studio.Net

Languages:

- VBScript
- JavaScript
- Transact - SQL
- MySQL (*language*)

Commercial Off-The-Shelf Solutions (COTS)

- Comercus Cart
- Cart32 Shopping Cart System for Windows
- CactuShop ASP Shopping Cart

Web Hosting:

- DiscountASP.Net

Microsoft SQL Server 2000 (SQL Server)

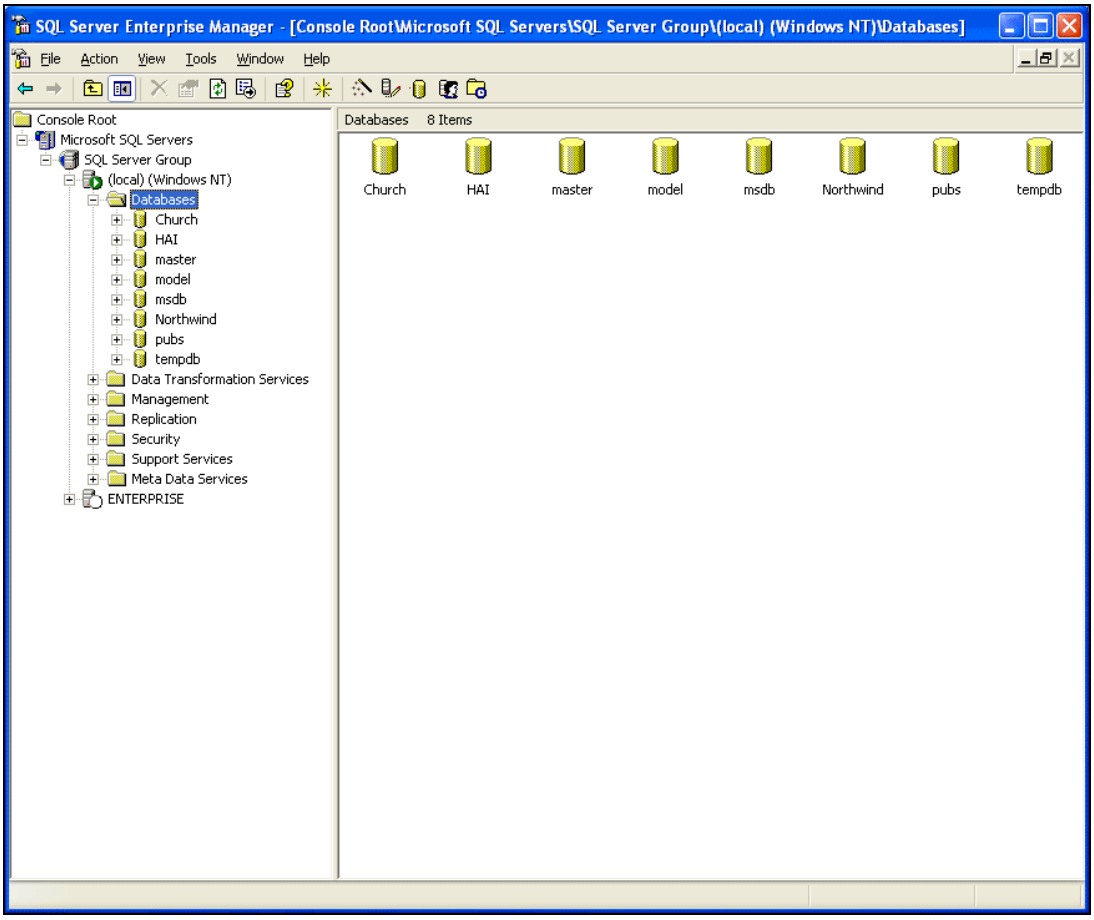
SQL Server is a widely used and reasonably scalable database solution that is fully integrate-able with FoxPro. Some of the features of SQL Server include:

- Enterprise Manager
    - Data Transformation Services (DTS)
    - Import and Export Data
    - Job Scheduling
    - Replication Services
    - Security (*administration*)
- Profiler

- Query Analyzer
- Security Network Utility

Although SQL Server offers several graphical tools to make maintenance and administration less complicated, it is not open source and free as there is no General Public License (GPL) however it is less expensive than other Relational Database Management Systems (RDBMS) such as Oracle, but again it is not nearly as powerful as Oracle.

An example of the SQL Server Enterprise Manager interface follows:



(Microsoft SQL Server)

Figure 7 – SQL Server Enterprise Manager

MySQL

MySQL is open source and free according to details outlined within their GPL. The worth of MySQL extends beyond the value of low cost and open source but includes benefits such as scalability to platforms unavailable to SQL Server. For example, SQL Server is not scalable to Sun Sparc architecture however MySQL is compatible with both .Net Framework on a Microsoft Windows machine as well as a Solaris Operating System (OS) on a Sun machine.

Some of the "free" features available through the MySQL site include:

- MySQL Administrator
- MySQL Query Browser
- MySQL System Tray Monitor

These tools provide similar features to SQL Server Enterprise Manager however they presume that the user of the tools has some experience manipulating and implementing databases. Compared to SQL Server, MySQL is not as a rule a "wizard-driven" collection of resources. An example of the MySQL Administrator Interface taken from the MySQL site follows:

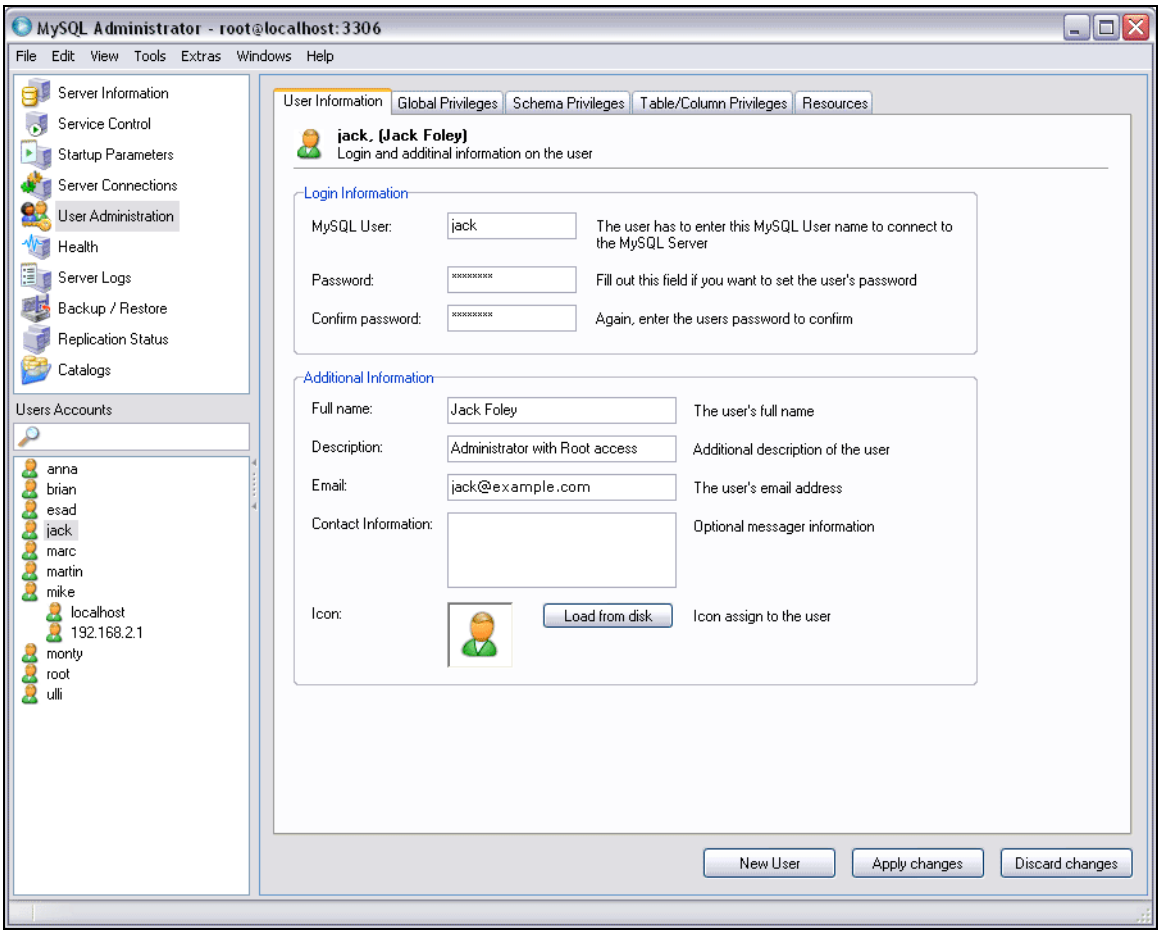


Figure 8 – MySQL Administrator

(MySQL Administrator)

Visual Studio and Visual Studio .Net

Visual Studio and Visual Studio .Net are tools well known to the developer and readily available with the suite of tools available as a resource. Using Visual Studio tools limits development opportunities to those compatible with a Microsoft Windows machine however there are many web hosting companies that support Visual Studio and the Visual Studio .Net Framework.

Languages

The languages selected are compatible with the SDLC methodologies that would work well in this type of project and include OO technology, they are VBScript and JavaScript. There are also many components available that support e-Commerce and encompass development strategies familiar to the developer.

The structured query languages proposed are also well known to the developer Visual Studio and Visual Studio.Net are tools well known to the developer and readily available with the suite of tools available as a resource. Using Visual Studio tools limits development opportunities to those compatible with a Microsoft Windows machine however there are many web hosting companies that support Visual Studio and the Visual Studio.Net Framework.

- VBScript
- JavaScript
- Transact – SQL (T-SQL)
- MySQL (*language*)

Commercial Off-The-Shelf Solutions (COTS)

Regarding developing the framework to implement the e-Commerce structure, one can begin the cycle of building and testing from scratch or search for a tested, competent solution and begin building on top of that. Considering the time involved in building an application from scratch and the scores of COTS solutions available, for this project it is better to focus on implementation than developing a solution when there are many capable applications available.

Affordable solutions that meet the requirements outlined above include:

- Comercus Cart
- Cart32 Shopping Cart System for Windows
- CactuShop ASP Shopping Cart

Comercus Cart

Comercus is an “asp shopping cart” application scalable to the selected databases outlined as probable candidates, MySQL and SQL Server. There are several price points to meet the needs of the church and there are price points to allow one to scale the application for other churches in the future. Payment options available from Comersus accommodate the variety of payment methods that may be requested by church members. These methods include:

- Support of a secure sockets layer (SSL) hosted from another site
- Payments by check
- Payments by credit card through reputable agencies such as PayPal
- Real time shipping through agencies/companies such as the Unites States Postal Service (USPS), Federal Express (FedEx) and the United Parcel Service (UPS)
- Security “encryption for sensitive data using a RC4” (*RC4 - variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation* (RC4/RC5/RC6))
- “HTML/Script injection attack verification in messages screen” (Comersus Sophisticated ASP Shopping Cart)

Some of the features include BackOffice reporting and templates (known as skins on the Comersus site). An example of a BackOffice report taken from the Comersus site shows a reporting example:

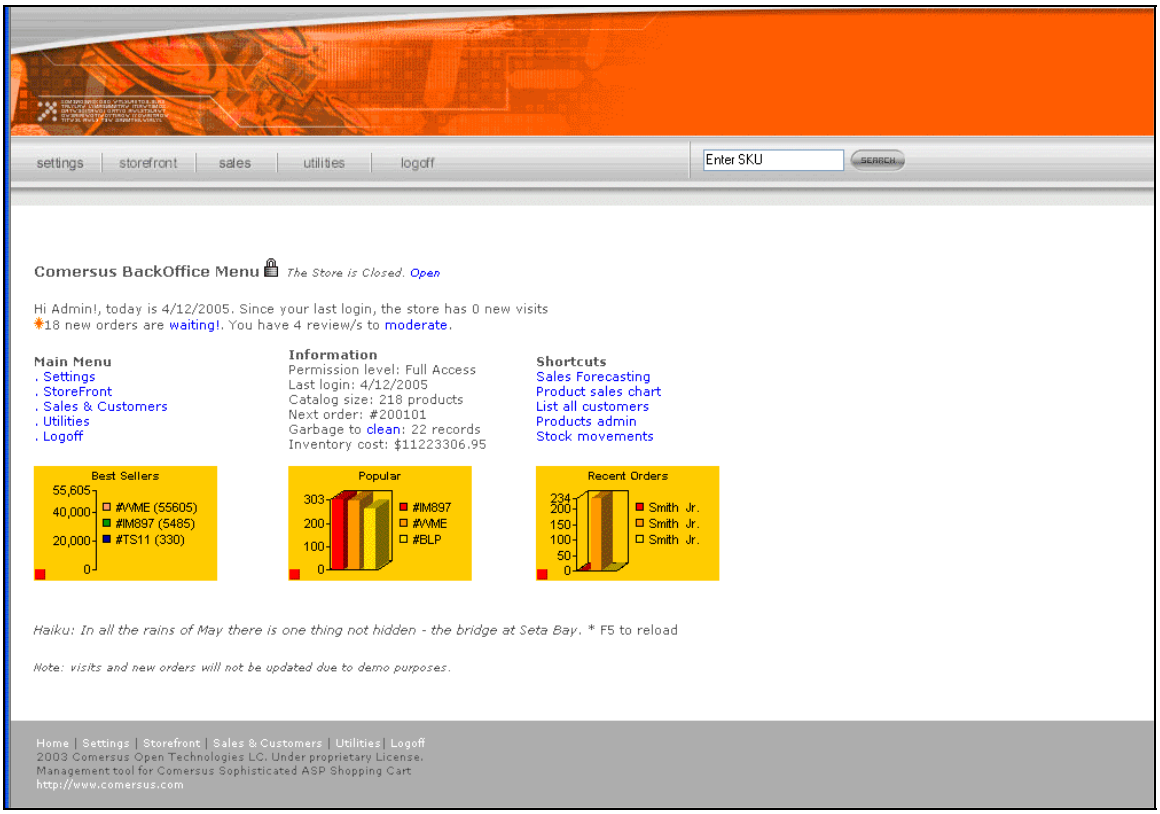


Figure 9 – Comersus BackOffice Demonstration

Another feature demonstrated on the Comersus site is the StoreFront. This component allows one to publish items for sale on the site, access to a "Members Area" that can be customized at some time according to requirements outlined at the church as well as other opportunities to communicate church events and other items available through e-Commerce. The following is a mock-up of the Bookstore StoreFront demonstration from the Comersus site.

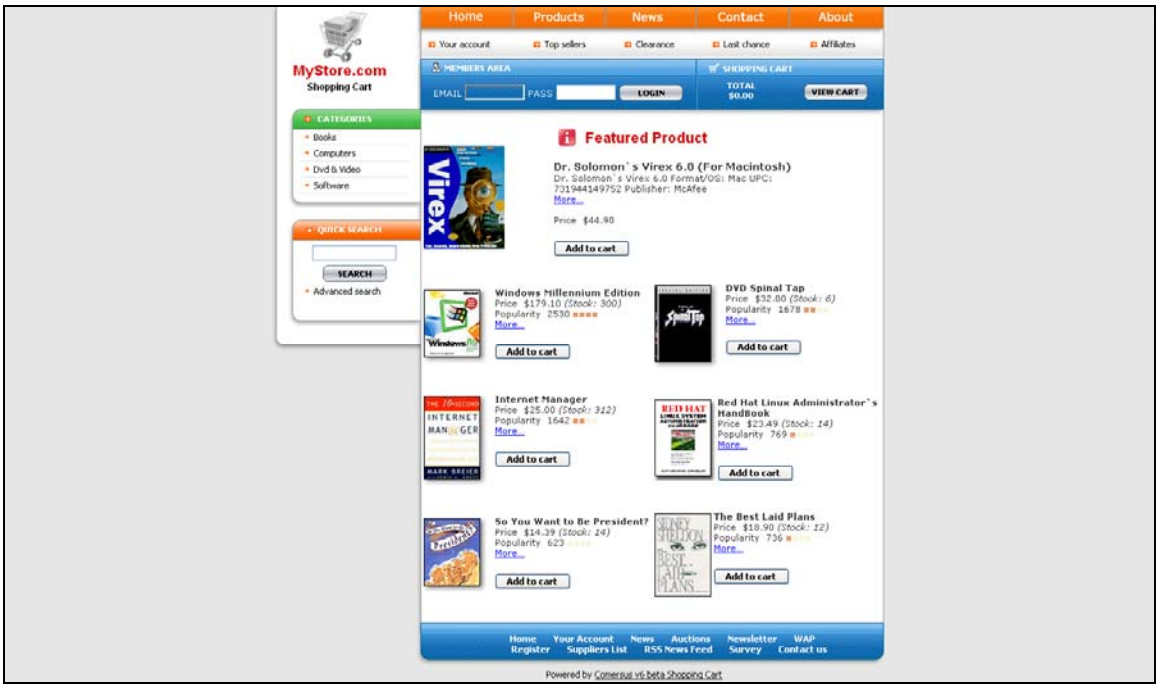


Figure 10 – Comersus StoreFront Demonstration

In addition to several development and user features Comersus offers a free “lite” version download, support packages and reseller opportunities.

The product is exceptionally constructed and feedback obtained from users attest to its soundness (The ASP Resource Index).

It is however essential to note that prior versions of Comersus lite are vulnerable to “Medium Risk” SQL Injection attacks:

> Comersus BackOffice Lite versions 6.0 and 6.01 are vulnerable to SQL injection, if the option pIndexVisitsCounter is set to -1. A remote attacker could exploit this vulnerability by sending a specially-crafted URL containing malicious SQL code, which would allow the attacker to add, modify or delete user information in the backend database.

Remedy:

Upgrade to the latest version of Comersus BackOffice Lite (6.02 or later), available from the Comersus Cart Download Web page. See References.

As a workaround, ensure the visitor logging (pIndexVisitsCounter = 0) is disabled (backoffice-lite-sql-injection (19013)).

Cart32 Shopping Cart System for Windows

Cart32 has many services including a “shopping cart” application scalable to the MySQL database. Although this service is not scalable to SQL Server it was worth a look as it offers development, database and hosting services which would allow a technically deficient organization some autonomy in maintaining their e-Commerce environment.

Some of the elements available from the Cart32 site include:

- Secure Public/Private Key Encryption
- Customize Shipping settings
- Supports Discounts
- Optional User Registration
- Optional Product Database
- Completely customize the look of the cart pages
- Accepts PayPal payments

(Cart32 Shopping Cart System for Windows)

One of the features demonstrated on the Cart32 site is available templates for publishing items for sale. An example of one of the templates taken from the Cart32 site follows.

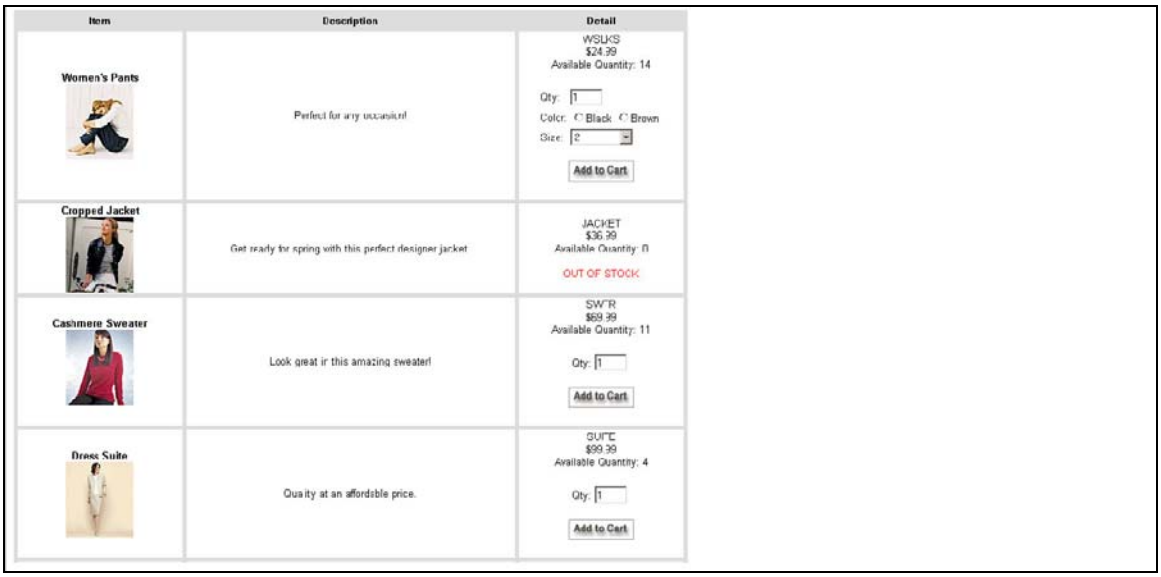


Figure 11 – Cart32 Store Manager Demonstration

The Cart32 product is well thought in that it provides resources for the novice developer however in reviewing feedback obtained from users, there is a flaw that without a doubt would cause anguish. The flaw outlined was a “secret password backdoor” (Multiple Vulnerabilities in Cart32 Shopping Cart).

CactuShop ASP Shopping Cart

“CactuShop is an ASP shopping cart software package which can be run on the vast majority of Windows web hosting with an Access, MS SQL or MySQL database used for storage. Full source code is supplied so almost any modifications or unique features that may be required could be added. Customizing the appearance of the cart is easier than for any other ASP shopping cart package - just edit a style sheet and HTML 'skin'. A lite demo version and documents are available as free downloads on this site” (CactuShop ASP Shopping Cart).

Thus CactuShop is scalable to the selected databases, MySQL and SQL Server. Through CactuShop, there are several price points to meet the needs of the church as well as a lite

version that is free to download. Payment options available from CactuShop include:

- 2CheckOut.com
- Authorize.Net
- PayPal
- SECPay
- SecureTrading
- VeriSign

(CactuShop ASP Shopping Cart)

Some of the features include “design skins”, design assistance and “UPS real-time shipping”. A detailed list taken from the site:

> Design
>
> CactuShop gives your store a clean look and professional design, with our unique skinning system, free skin downloads, language strings and dozens of configuration settings for customizing the layout.
>
> Customers
>
> Your customers will have everything at their disposal to browse and purchase items from your store with ease, including the basket, speed ordering, order tracking, mailing list, save and recover baskets and customer wish lists.
>
> Backend
>
> CactuShop's backend user interface gives you complete control over your store. It provides the ability to look up and set the progress of orders, manage the product catalogue, view/edit your customers and affiliates, send mail, modify site configuration and monitor page statistics.

Catalogue

CactuShop allows a great deal of flexibility in setting up products and categories. The number of products and categories supported is virtually unlimited and you can create multiple levels of categories and subcategories and feature products is multiple categories if you wish. Add to this stock control, electronic downloads support for large and normal images and you have a solution that should fit most requirements.

System & Support

CactuShop is easy to set up. It runs efficiently with an MS Access, MS SQL Server or MySQL database. We also provide free tools for data import and upgrading, as well as an online knowledgebase.

Regional Setup

CactuShop can run on stores all over the world. Multiple languages and multiple currency support means you can set up your one store to target more than one buying country.

Payment

CactuShop supports many of the most popular online payment gateways so you can securely accept credit card payments. You can also set the store to use offline methods (such as cheque or postal order) or accept card details through your own secure area.

Shipping & Tax

Shipping can be calculated based on either the weight or price of an order,

or you can look up real time shipping rates from UPS.

Email

CactuShop supports all the most popular methods for sending email from ASP pages.

(CactuShop ASP Shopping Cart)

CactuShop Ltd. is physically located in the United Kingdom and pricing information is provided in pounds rather than dollars however the software product is indeed stellar. Regarding the reliability of the product, the site provides testimonials from users. However in researching further there is a known “High Risk” backdoor noted on the Internet Security Systems Inc, site with the “lite” version of CactuShop that must be considered before selecting this solution (cactushoplite-backdoor (15063)).

Web Hosting

DiscountASP.Net

DiscountASP.Net is a hosting service that provides:

- General Hosting Features
- Domain Name Features
- Email Features
- .Net Framework
- Database Support and Features
- ASP/.Net Components
- Authoring Tools

In addition to offering many features and components, DiscountASP.Net is economical

and easy to use. An example of an ease of use is feature is the availability of connection string source for SQL Server databases, thus the developer resource(s) do not have to create a data source name (DSN), this resource dynamically created by DiscountASP.Net. Another benefit is backup services are provided, thus the church would have very little risk of losing information and would not have to invest in a backup solution.

One objection to using this hosting service is that the developer is locked into a Windows machine as the DiscountASP.Net datacenter is committed to Windows and the .Net Framework (DiscountASP.Net).

3.0 Methodology

Overview of Selected Methods

The decision to plan this project using the selected strategies is a result of:

- Understanding the customer (the church membership)
- Determining solutions that would not likely be threatening
- Past experience prototyping and developing applications
- Past experience with MySQL and SQL Server RDBMS
- Researching various web sites
- Reviewing SDLC methodologies not necessarily familiar (i.e., Evolutionary) but interesting and worth looking into

As a result, the following approaches were selected as a result of the conducted research.

Phase I – Systems Analysis Phase

Software Development Life Cycle

The Spiral Model was selected since it incorporates the Prototyping and Waterfall methods. Prototyping along the way will provide the necessary flexibility to provide a "sneak preview" to the church decision makers while the Waterfall approach will present the necessary structure.

A mockup of the structure of the Waterfall method commented to fit this project is as follows:

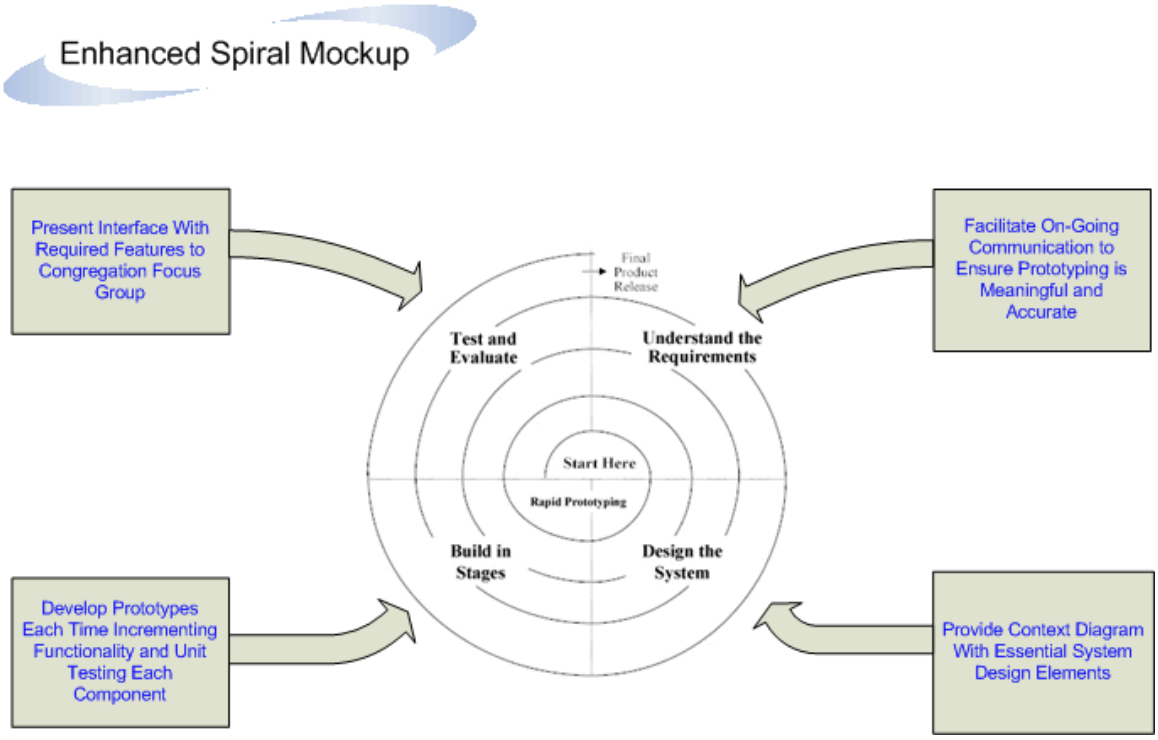


Figure 12 – Enhanced Spiral Mockup

While discussing the elements of this project with the Pastor, it was evident that he would benefit from seeing the application unfold in mini steps thus to ensure the first release would meet his needs and future releases could be accomplished without a complete redesign.

Another consideration in selecting this method is the overall lack of technology resources at the church. Developing an application using this system is a resourceful approach to gradually obtain the confidence of the members not familiar with not only e-Commerce, but web technology itself. For example many church members are familiar with the Internet but are skeptical of e-Commerce due to reports of stolen personal data however in order to overcome come this challenge one must research case studies where Project Teams have had to overcome this doubt.

System Requirements

In order to describe the proposed application precisely prior to beginning the first prototype, the following Unified Modeling Language (UML) diagrams are employed to illustrate system requirements and enhance understanding regarding how users will interact.

Use Cases

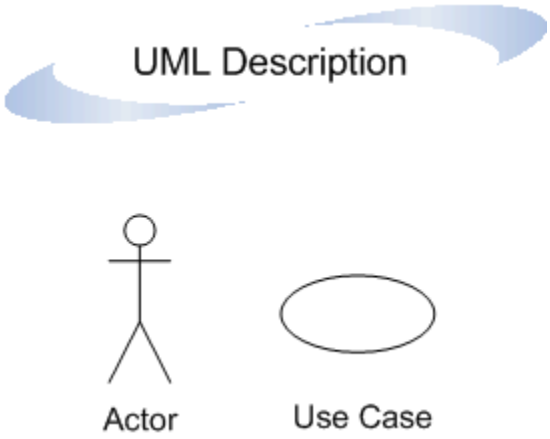


Figure 13 – Unified Modeling Language Description

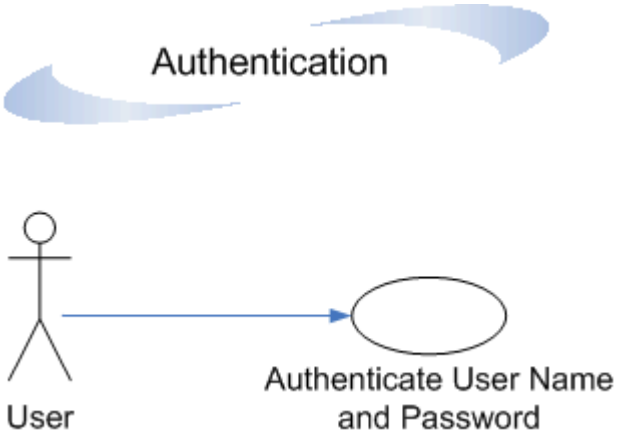


Figure 14 – Use Case 001 Authenticate User ID and Password

Table 1 – Use Case 001 Authenticate User ID and Password

Use Case ID:	UC_001		
Use Case Name:	**Authenticate User ID and Password**		
Created By:	Montez A. Toney	Last Updated By:	Montez A. Toney
Date Created:	April 14, 2005	Date Last Updated:	April 14, 2005

Actors:	e-Commerce Database Application Systems Administrator
Description:	This form is the default web page for the System Administrator. It is used to mutually validate the user name and password.
Preconditions:	1. Data elements (user name and password) must be in the user table within the e-Commerce database schema. 2. Both the user name and password must be entered using the proper format. 3. Password is less than 90 days old.
Post Conditions:	1. The e-Commerce database reporting page may become navigable and the user's browser will automatically redirect to this page.
Normal Flow:	1. The e-Commerce database reporting page will appear so that the user can begin generating reports.
Exceptions:	1. Error in data entry. 2. Web interface error. 3. User does not proceed to the e-Commerce database reporting page. 4. Password is 90 days old or greater. 5. Web interface error redirecting user to update password. 6. Several (TBD) incorrect login attempts will trigger an

	email to the System Administrator.
Includes:	1. Dynamic last updated message. 2. User name and welcome message upon authentication.
Frequency of Use:	1. TBD by church focus group.
Business Rules:	1. Processed anytime a user navigates to the e-Commerce database reporting site. 2. Processed if a user's web browser has been inactive less than 30 minutes. 3. User must update password every 90 days. 4. Authentication and application will support concurrent (defined as two or more) logins.

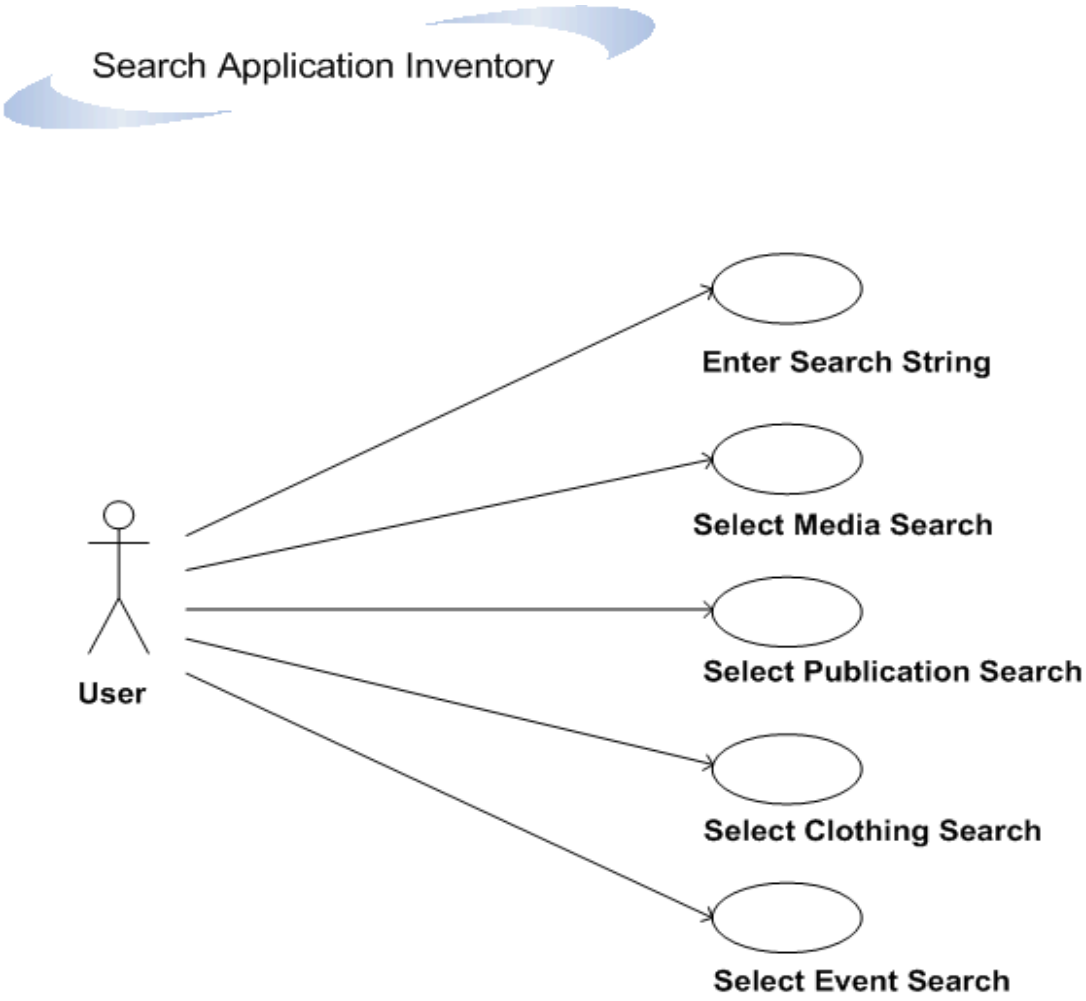


Figure 15 – Use Case 002 Search Application Inventory

Table 2 – Use Case 002 Search e-Commerce Database Application Inventory

Use Case ID:	UC_002		
Use Case Name:	**Search e-Commerce Database Application Inventory**		
Created By:	Montez A. Toney	Last Updated By:	Montez A. Toney
Date Created:	April 14, 2005	Date Last Updated:	April 14, 2005

Actors:	e-Commerce Database Application User
Description:	This form is the subsequent page to navigating to the available

	inventory page.
Preconditions:	1. Successful navigation to the e-Commerce database application.
Post Conditions:	1. Data elements “LIKE” entries on e-Commerce database application Search Options page will appear.
Normal Flow:	1. The user will enter or select one search criterion. 2. The e-Commerce database application will return matching records.
Exceptions:	1. User does not enter or select any search criteria. 2. Web interface error. 3. Error in data entry. 4. Web interface error. 5. No records exist matching (“LIKE”) entry criteria. 6. Web interface error.
Includes:	1. Dynamic last updated message. 2. User name and welcome message throughout navigation experience.
Frequency of Use:	1. Processed multiple times per individuals. Usage is determined by whether suitable results are acquired.
Business Rules:	1. May be processed anytime a user successfully authenticates user name and password mutually. 2. Processed if a user’s web browser has been inactive less than 30 minutes. 3. User must enter or select at least one search element in order for the e-Commerce database application to query data.

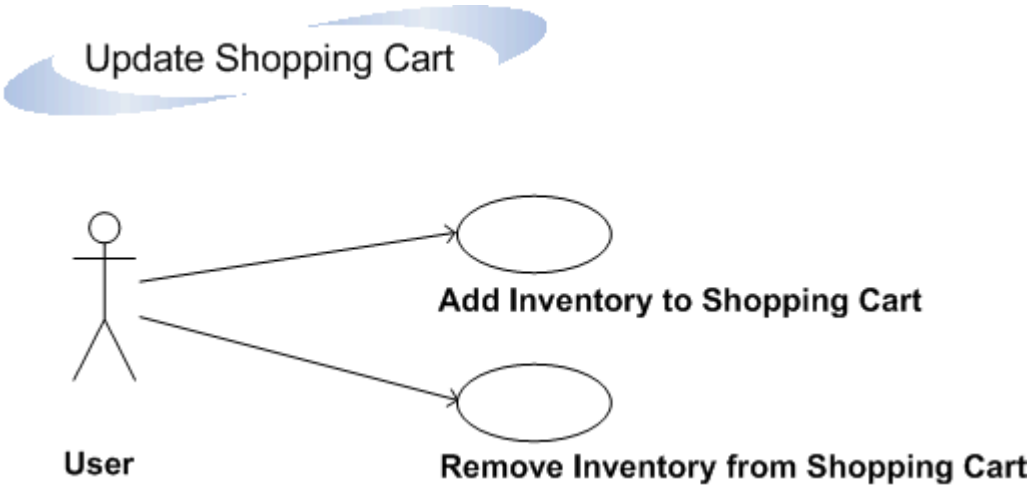


Figure 16 – Use Case 003 Update Shopping Cart

Table 3 – Use Case 003 Update Shopping Cart

Use Case ID:	UC_003		
Use Case Name:	**Update Shopping Cart**		
Created By:	Montez A. Toney	Last Updated By:	Montez A. Toney
Date Created:	April 14, 2005	Date Last Updated:	April 14, 2005

Actors:	e-Commerce Database Application User
Description:	This option is the subsequent action prior to selecting the option to navigate to the checkout.
Preconditions:	1. Item(s) must be added to the shopping cart.
Post Conditions:	1. Item(s) will be added to the shopping cart. 2. Items will be removed from the shopping cart.
Normal Flow:	1. The user will select inventory to be added or removed from the shopping cart.
Exceptions:	1. User does not add any inventory to the shopping cart. 2. Web interface error.

	3. Error in data entry. 4. Web interface error. 5. User cancels transaction. 6. Web interface error.
Includes:	1. Dynamic last updated message. 2. Welcome message throughout navigation experience.
Frequency of Use:	1. Processed multiple times per individuals. Usage is determined by whether suitable results are acquired.
Business Rules:	1. May be processed anytime a user selects option to add or remove inventory. 2. Processed if a user's web browser has been inactive less than 30 minutes. 3. User must enter or select at least one search element in order for the e-Commerce database application to query inventory. 4. User must have at least one item in shopping cart in order to remove an item.

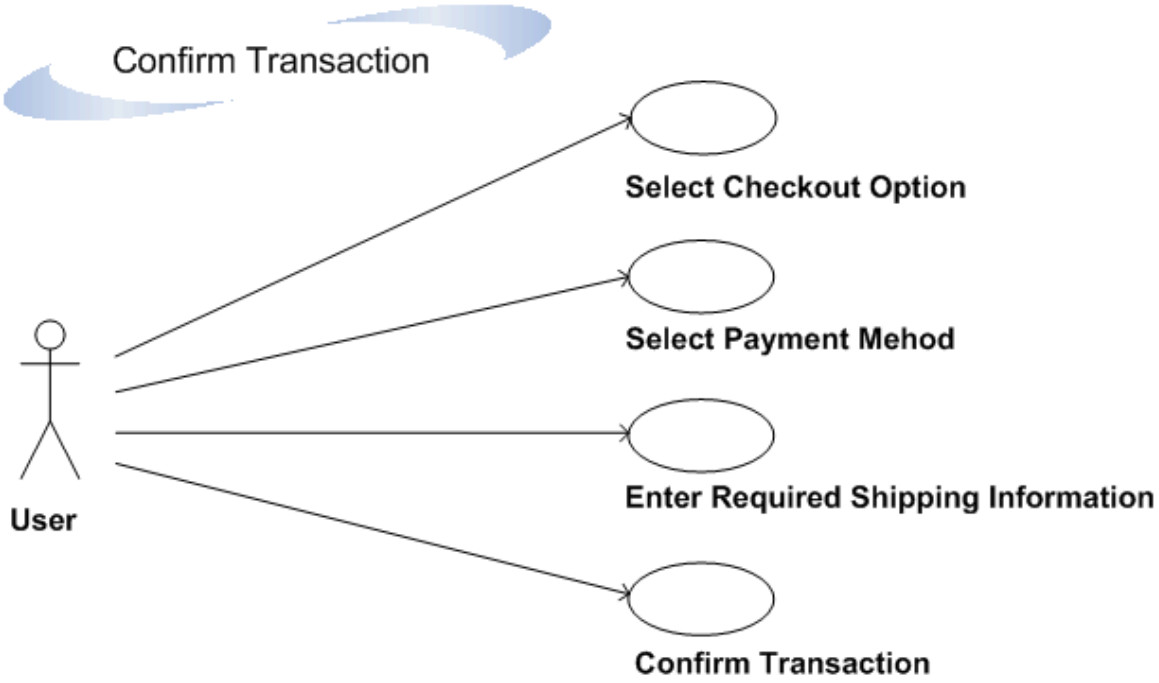


Figure 17 – Use Case 004 Confirm e-Commerce Transaction

Table 4 – Use Case 004 Confirm e-Commerce Transaction

Use Case ID:	UC_004		
Use Case Name:	**Confirm e-Commerce Database Application Transaction**		
Created By:	Montez A. Toney	Last Updated By:	Montez A. Toney
Date Created:	April 14, 2005	Date Last Updated:	April 14, 2005

Actors:	e-Commerce Database Application User
Description:	This form is the subsequent page to selecting the option to navigate to the checkout.
Preconditions:	1. Successful shopping cart update with content(s) added.
Post Conditions:	1. Amount owed will be calculated and added to the checkout total.
Normal Flow:	1. The user will enter or select one payment method.

	2. The user will update the ship to name if applicable. 3. The user will update the shipping address is applicable. 4. Confirm transaction.
Exceptions:	1. User does not add any inventory to the shopping cart. 2. Web interface error. 3. Error in data entry. 4. Web interface error. 5. No records exist matching (“LIKE”) entry criteria. 6. Web interface error. 7. User cancels transaction. 8. Web interface error. 9. User does not enter required shipping information. 10. Web interface error.
Includes:	1. Dynamic last updated message. 2. Welcome message throughout navigation experience.
Frequency of Use:	1. Processed multiple times per individuals. Usage is determined by whether suitable results are acquired.
Business Rules:	1. May be processed anytime a user successfully selects inventory. 2. Processed if a user’s web browser has been inactive less than 30 minutes. 3. User must enter or select at least one search element in order for the e-Commerce database application to query inventory.

Technical Requirements

In synchronization with the Use Cases, the following researched components will be utilized in order to build the e-Commerce database application:

- SQL Server as the RDBMS (Note: MySQL is available also however it is not the first choice)
- Integrated Development Environment (IDE) provided through Visual Studio
- Development languages include VBScript, JavaScript and Transact-SQL
- Shopping cart by means of Comercus Cart
- Web hosting through DiscountASP.Net
- Users must employ either Internet Explorer 6.0 or higher OR Netscape Navigator 7.0 or higher

Phase II – Design Phase

Given that there is no existing application in place, there is no obstacle to overcome in correcting it. Thus the application design can be modeled after a solution in existence and likewise working well elsewhere.

The e-Commerce database application will provide a collection of items for purchase as determined by the church. Part of providing the system will involve reporting resources so that the church can accurately track transactions taking place on the site.

In order to begin describing the design, a Context Diagram outlining the flow of the application is illustrated:

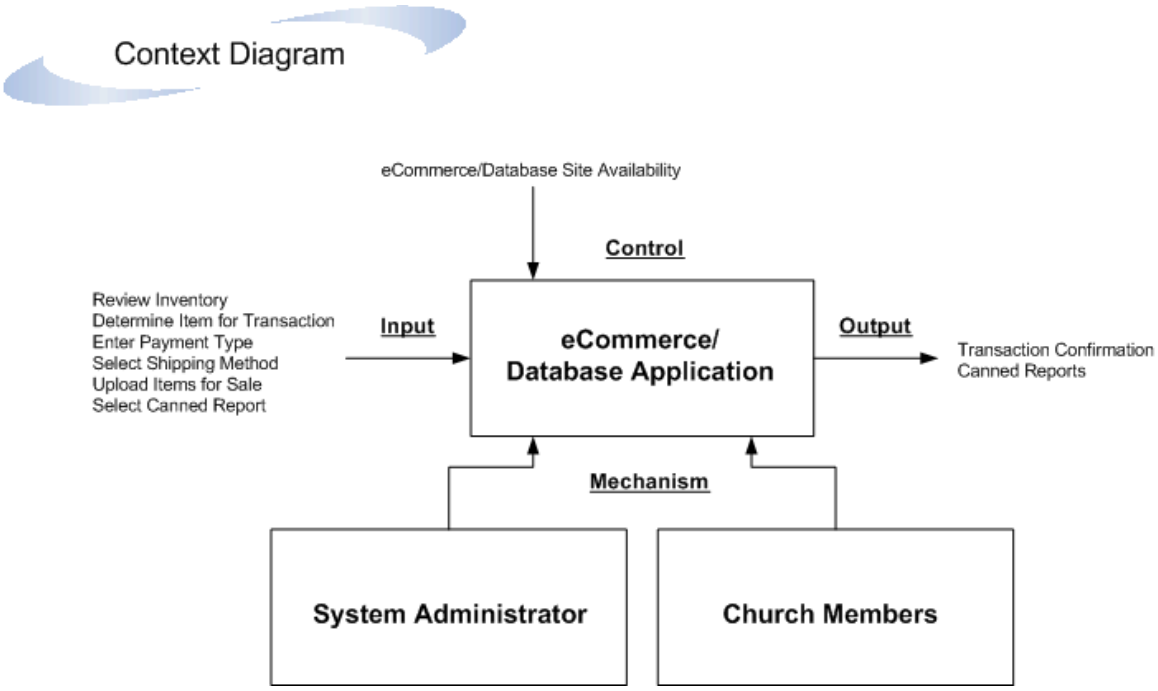


Figure 18 – Projected Context Diagram

System Architecture

The e-Commerce database solution is fulfilled within a 3-tier infrastructure. The Client Tier consists of a web browser running within a browser window keeping in mind compatibility with both Netscape and Internet Explorer. The graphical user interface (GUI) will consist of HTML, DHTML and JavaScript elements as supported by the Client Tier. The Application Tier contains an IIS web server and application server. The Database Tier encompasses the data sources.

Following the requirements, the architecture outlined follows the services provided through the selected web hosting company, DiscountASP.Net.

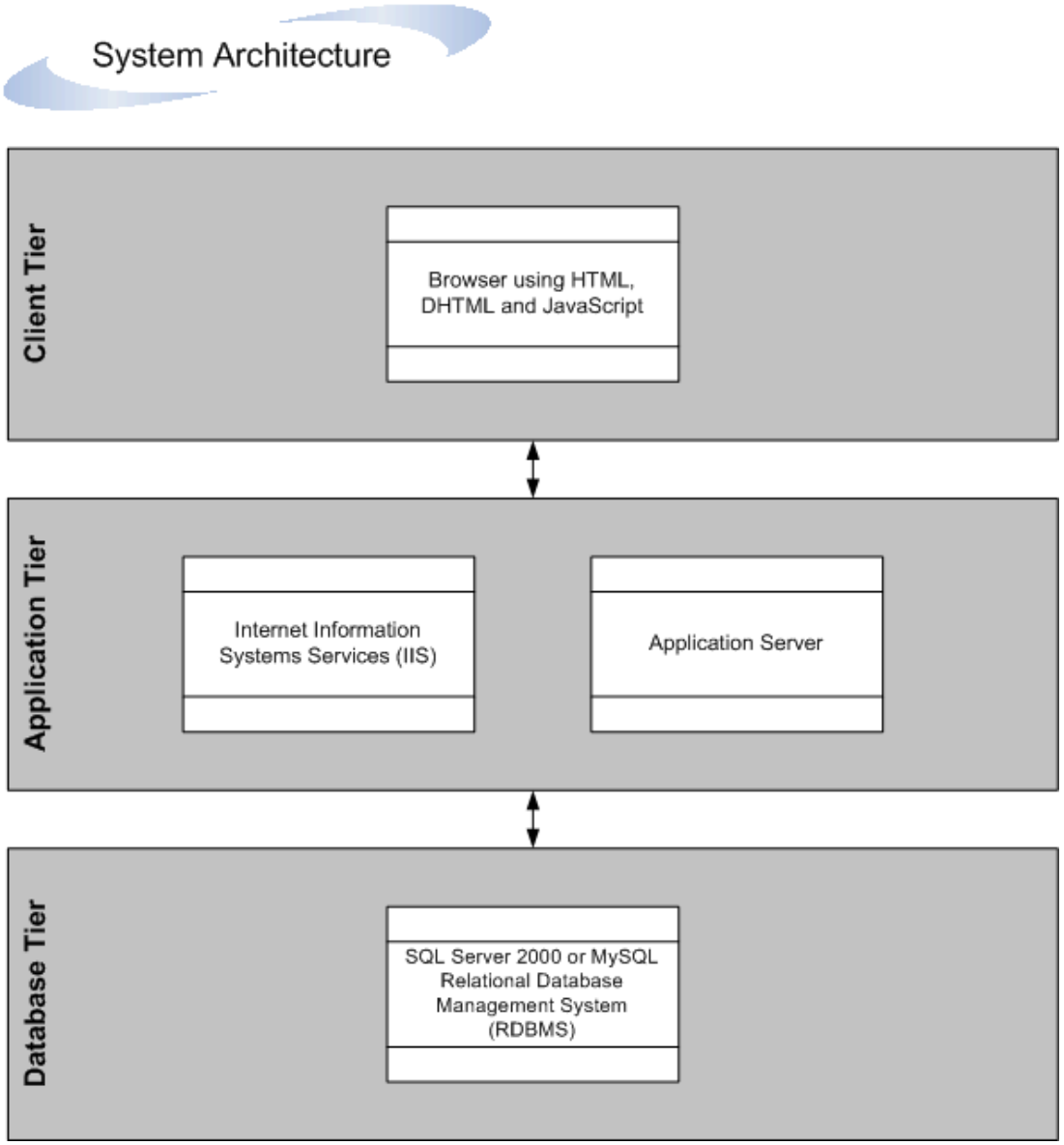


Figure 19 – Projected System Architecture

Phase III – Development Phase

Developing the e-Commerce database application will be a result of receiving adequate buy-in from church members regarding the system Use Cases. Through collecting feedback and likewise updating the prototype, mock-ups of the projected development result are according to the "Agile Model Driven Development" method described by Scott W. Ambler.

Initial Requirements Modeling and Initial Architectural Modeling

Within the first week of beginning the project, develop somewhat of a working prototype. This will include creating a framework for the database, i.e. entity relationship diagram (ERD), creating tables and creating simulated data and loading it into the database. Also within the first week of creating a prototype, develop a working interface. Since a COTS product (Comercus) will be used, implementing this feature will most involve discovering the nuances of the product and utilizing the features that meet the project requirements and determines if any features that need to be developed from scratch by the developer.

To demonstrate the prototype, secure service through the selected hosting company, DiscountASP.Net and move forward with the presentation.

During the presentation it is critical that feedback is gathered as well as asking the difficult questions to making the project successful. Some of these would be:

What went well?
What could be improved for the next prototype presentation?

After these questions are answered, the information must if it is within the scope of the project be updated within requirements, i.e. Use Cases, Context Diagram etc., and presented again to the Focus Group assigned to follow the project.

As noted by Ambler, it is recommended that this portion of the project not extend beyond two weeks as it could indicate that the prototype is being "over developed" or important collaboration is being missed between the Project Team and the Focus Group.

Model Storming

Modeling Storming is the process of gathering an abbreviated segment of the Project Team as well as the customer receiving the deliverable. In the case f this project the customer is the church focus group. Through this process, components of the project requirements are mentioned and mocked-up quickly on usually a white board.

Although these meeting are brief, usually less than 30 minutes much progress has occurred. Developers are provided with documentation and usually a UML diagram is mocked up so that complex programming tasks are well thought out prior to developing code without much of a roadmap.

Implementation

Following Model Storming and developing functionality is the Implementation process. Ambler discusses using a role referred to as a "Generalizing Specialist, someone who can write code and also model" an application. As the project progresses, having resources with only one skill is not the most efficient manner of completing a task, thus selecting a multi-faceted staff will ensure continuity with development as well as resources that can implement the project.

Ambler models this process very well in an easy to follow diagram. Note the diagram appears to have other pages behind it. These pages suggest the "cycles" or iterations of the project through Prototyping, Model Storming and Implementation.

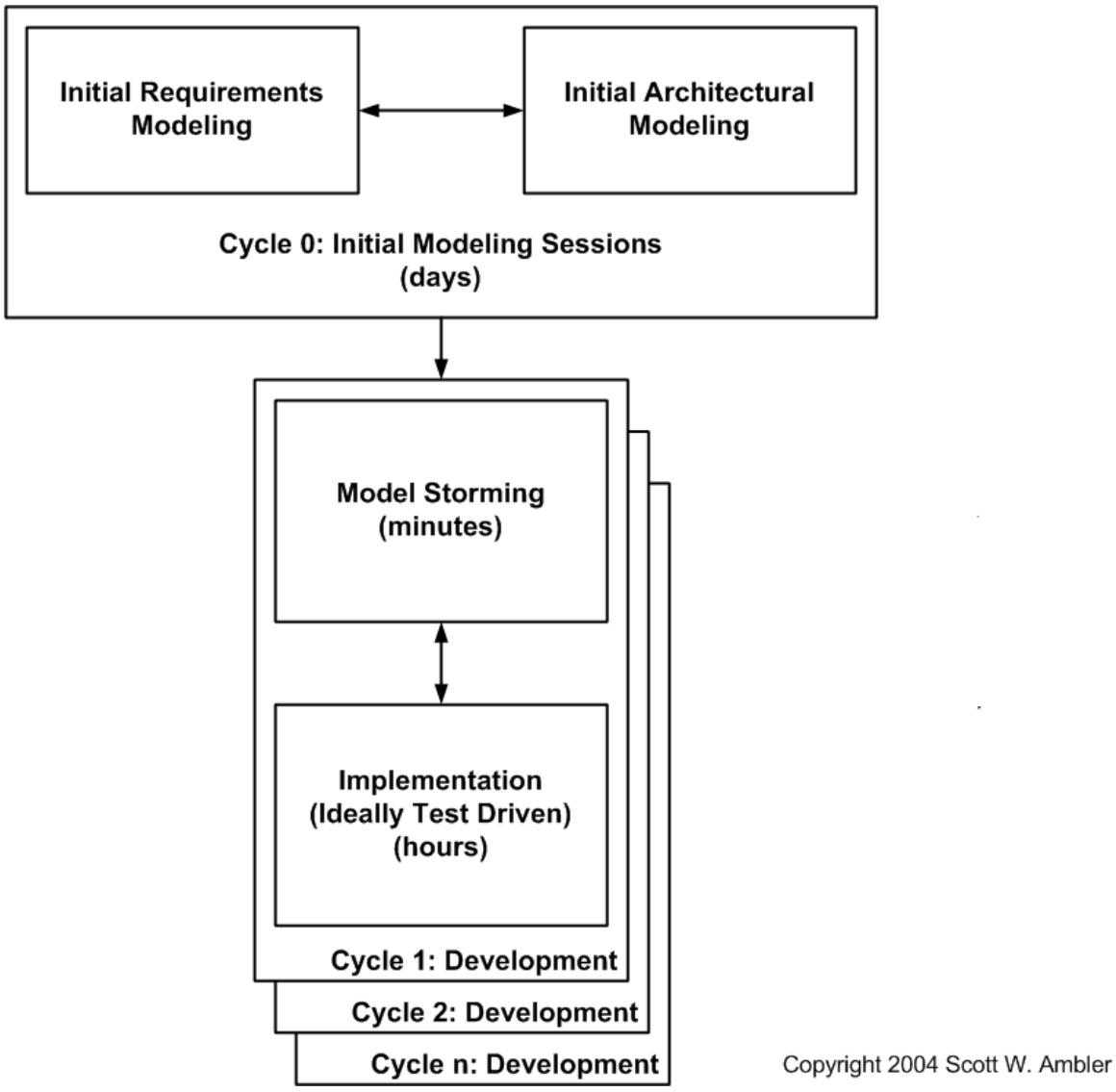


Figure 20 – Agile Model Driven Development (AMDD)

(Agile Model Driven Development)

Phase IV – Testing Phase

Within the scope of an accepted test strategy, an alpha and beta test will be conducted to support validation of the e-Commerce database application.

Testing will follow after the agreed upon requirements are fulfilled and the project is deemed ready to prepare for implementation. This process will include writing formal

test plans using the application requirements. The test plan will include the following elements:

- Title of the segment to be tested
- Corresponding requirement
- Name of tester
- Date the segment was tested
- Whether the item passed or failed the test element

Alpha Testing

The alpha test will be conducted by the developer and structure will include:

- Critically reviewing each requirement and ensuring the correct functionality is in place within the prototype and is error free
- Selecting a segment of the focus group to review the final prototype and conduct light UAT for the purpose of gathering feedback that is not immediately obvious to the developer

Beta Testing

The beta test will be conducted by the members of the focus group designated to conduct testing and will present the fully developed application in the form of a prototype. The beta test environment will be identical to the projected production environment and will include:

- The focus group will be provided with a UAT Test Plan and must likewise conduct the test following each step to ensure that the application is working according to the accepted requirements.
- The application will be fully functional on hardware designated by the focus group and all beta testing should be conducted on it to ensure accurate test results.

Once test results are collected, they will be submitted for review to determine whether the application is ready to be promoted to the Implementation Phase or whether any areas of concern can be resolved within the time allocated within the project plan.

Phase V – Implementation Phase

The Systems Analysis, Design, Development and Testing Phase are all part of Implementation as without completing these critical elements, putting an application in place could be disastrous.

Thus this phase is a culmination of the success of the aforementioned project milestones. The final implementation will be acceptance of the delivered application, resolution of any testing errors or concerns as well as providing project documentation in the event that the project will be updated or modified sometime in the future.

Phase VI – Training and Maintenance Phase

Training

Regarding training users to receive the benefit of e-Commerce, teaching will be conducted through the following three strategies:

<u>Conduct Demonstration Sessions at Church</u>

The demonstrations will be no more than 30 minutes, 20 minutes to display the application functionality and 10 minutes to answer questions.

Through the process of sharing this idea with various church members, similar questions crop up. The main concern is whether of not it is safe to submit a credit card number

over the web. The answer to this question of course is it depends. There are measures that should be taken to make certain that checks and balances are in place to protect the consumer. The following article is taken from the Verizon web site to demonstrate to their customers that using credit card information over the web can be especially safe:

Shopping online has become very safe. However, there are a few tips you should consider anytime you share your personal information online.

- Make sure the Web site is secure. Secure Web sites use "Secure Sockets Layer," an encrypted protocol that protects confidential user information. You can identify an SSL-protected page in one of two ways:

1. SSL-protected-Web pages will display a closed-lock icon on the status bar at the bottom of the browser window:

Table 5 –Secure Sockets Layer Status

Browser	Secure Web Page SSL Layer
Microsoft Internet Explorer	 Closed lock
Netscape Navigator	 Closed lock

2. The URL of an SSL-protected- Web page will begin with "https" rather than "http." For example: http://www.verizon.com would be an insecure page, while https://www.verizon.com would be SSL-

encrypted and could be considered safe for credit card transactions.

- Check your credit record on a regular basis. You should request a credit report from each of the three major credit bureaus at least once each year and verify that all the information is correct (Verizon Online - Help & Support - Is it safe to use my credit card online?)

Develop Detailed Documentation

Documentation will be created to support the application details however it will be brief as to ensure it is less likely to intimidate readers. It will include screen shots and follow a structure organized in the same order as the demonstration.

Develop a Handout

A one-page handout will be created to emphasize major highlights and frequently asked questions (FAQ).

This method of communicating is being applied to accommodate various learning styles and ensure distributing information to the largest possible segment of the congregation.

Maintenance

Application maintenance will be conducted by the designated focus group member and will include:

- Ensuring new data elements are loaded into the database
- Minor website updates (text changes)
- Making certain the DiscountASP.Net invoices are paid

- Periodically checking the site for broken links and other error

4.0 Project History

Why an e-Commerce Application?

The decision to develop an e-Commerce application was for the writer of this work to give something back to the church to utilize and learn from as well.

Congregations are sometimes made up of a diverse group of cultures and professional backgrounds however it is not often that a member steps up to present an alternative to the status quo. Church websites are becoming very common however few churches have actually taken the leap to a real e-Commerce site.

In searching for examples to compare (within the author's denomination), any pages that included commerce were order forms, phone numbers and e-mail addresses. No real time inventory or payment systems were sited.

Confirming an Interest

After determining an opportunity to introduce a new technology dimension, the church pastor was consulted to establish whether the interest was mutual.

The response from the pastoral staff was overwhelmingly positive. Innovative ideas poured regarding e-Commerce and web based technologies however the scope of this project was limited to something reasonable to implement with a few months.

In addition to the pastoral staff, there is a Technical Assistant with an interest in gaining exposure to more database and programming skills. Thus, there is someone in place to provide application support whether or not the application developer is available.

The Project Team

All project roles other than the church focus group are fulfilled by the writer of this work. This does present some challenges as well as benefits however overall this undertaking has been quite a learning experience.

Project Challenges

Although there is a Technical Assistant on the church staff, this proved to be more of a hindrance to fulfilling the initial requirements. For example, at the inception of the project the pastor asked for an ongoing meeting weekly. During these meetings requirements would be refined and prototypes would be shared.

At the same time the Technical Assistant asked the author privately to share all details of the pastoral meetings which was obliged. However upon sharing the information the author was asked to discontinue the weekly meetings with the pastoral staff rather they would be conducted by the Technical Assistant and the author would subsequently be informed regarding how to move forward with the project.

This was indeed a setback and somewhat of a roadblock to accomplishing the task. A decision had to be made regarding whether the author would share this information with the pastor or follow the instructions of the Technical Assistant.

The author chose not to push this sticky situation and instead pondered strategies to complete the project with minimal risk to exposing the Technical Assistant.

Why This Work Took So Long

In addition to uncovering the bottleneck attempt by the Technical Assistant, the author

moved from Florida to Maryland. During the move somehow the computer hardware used to write this work was packed.

Upon completing the move, adjusting to the new culture, commuting over four hours per day and adjusting to the cold made it difficult to pick up where one left off.

Also, the longer the author waited to begin working on this project again, the more difficult it was to commence.

Really Delivering

Regardless of the challenges with this project it will indeed be developed and implemented. Upon approval of this document, the pastor at St. Paul A.M.E. church will be contacted and a prototype will follow.

This project came about from a desire to share knowledge and technology and although an individual (Technical Assistant) is not receptive, the work will not be stopped.

5.0 Lessons Learned and Next Evolution of the Project

People are People

Just because one wants to give something away does not mean that everyone really wants it. Also just because people are agreeing with something at church does not mean those are their true feelings.

A project developed for church is no different than a project developed at work. In many cases there are more challenges as at places of employment there are usually processes and procedures in place to ensure concerns are escalated properly. Also, the mission of the church is to save souls, not employ technical solutions.

Forward Thinking - People are People

Future releases of this project will consider the reality that the author's excitement about a deliverable may not be shared by all and thus the initial statement of work will provide for communication with the pastoral staff as well as sharing essential feedback and suggestions regarding how to move forward.

Would the Author do This Again?

Yes, but different.

Mostly the author learned to keep plugging along no matter what the problem to overcome. The author was indeed at a serious risk not complete this project due to not being assertive enough to proceed under the given circumstances.

Also the longer one waits the harder it is to continue. October 2004 the author attempted to quit her job, however was asked to continue at home on a part-time basis. Part-time became full-time and full-time became overtime.

During March 2005 the author's employer lost their funding for three weeks. It was during this time that serious dedication to this project took place and the result is completing this document.

Next Evolution for the Project

First and foremost deliver a prototype to St Paul A.M.E. church.

Following delivering a prototype, the next step for this project is to share it with other churches. Providing an e-Commerce solution is more than providing a working solution but perhaps getting others interested, exposure to technology. Throughout this undertaking there have been many to express an interest, but they lack a vehicle to help move them along.

Hopefully this project will be implemented successfully at many churches and likewise present a learning opportunity as well.

Glossary

ASP	Active Server Pages
CSS	Cascading Style Sheet
COTS	Commercial Off The Shelf
DHTML	Dynamic Hyper Text Markup Language
DSN	Data Source Name
DTS	Data Transformation Services
e-Commerce	Electronic Commerce
e-mail	Electronic Mail
ERD	Entity Relationship Diagram
FedEx	Federal Express
GPL	General Public License
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
IIS	Internet Information Server
ISP	Internet Service Provider
IT	Information Technology
MS SQL	Microsoft Structured Query Language
OO	Object Oriented
OS	Operating System
PMBOK	Project Management Body of Knowledge
RAD	Rapid Application Development
RC4/RC5/RC6	Run Command
RDBMS	Relational Database Management System

SDLC	System Development Life Cycle
SSL	Secure Sockets Layer
SQL	Structured Query Language
UAT	User Acceptance Test
UML	Unified Modeling Language
UPS	United Parcel Service
URL	Uniform Resource Locator
USPS	United States Postal Service
VB	Visual Basic

Works Cited

2005. DiscountASP.Net. 13 Apr. 2005 <http://www.discountasp.net/features.aspx>.

2005. Servant PC Resources Inc. 07 Apr. 2005 <http://www.servantpc.com/>.

Ed. Matt Stephens. 2005. Software Reality. 06 Apr. 2005 <http://www.softwarereality.com/design/agileiconix.jsp>.

Ed. Scott W. Ambler. 2005. Ambysoft Inc. 15 Apr. 2005 <http://www.agilemodeling.com/essays/amdd.htm>.

Ed. Anthony Barratta. Apr. 2000. Cerberus Information Security, Ltd. 12 Apr. 2005 <http://lists.evolt.org/archive/Week-of-Mon-20000424/100179.html>.

2005. InfoGenius Inc. 12 Feb. 2005 <http://www.aspin.com/func/review-hot?id=1804110&rid=&pg=1&order=desc>.

2005. Cactusoft Ltd. 12 Apr. 2005 <http://www.cactushop.com/>.

Feb. 2004. Internet Security Systems, Inc. 13 Apr. 2005 <http://xforce.iss.net/xforce/xfdb/15063 >.

2005. 12 Apr. 2005 <http://www.cart32.com/>.

Jan. 2005. Internet Security Systems, Inc. 13 Apr. 2005 <http://xforce.iss.net/xforce/xfdb/19013>.

2004. Comersus Open Technologies. 12 Apr. 2005 <http://www.comersus.com/store/ppremium.html>.

Jirachiefpattana, Waraporn. "The Impacts of Thai Culture on Executive Information Systems Developmenta." International Journal of Computer and Engineering Management (1997): 06 Apr. 2005 <http://www.journal.au.edu/ijcem/may97/article4.html>.

2005. Microsoft. 07 Apr. 2005 <http://www.microsoft.com/sql/default.mspx>.

Nov. 2000. Xato Network Security, Inc. 12 Apr. 2005 <http://www.cgisecurity.com/archive/shop/multicart32.html>.

2005. internet.com. 07 Apr. 2005
<http://www.webopedia.com/TERM/M/MySQL.html>.

2005. MySQL AB. 07 Apr. 2005 <http://www.mysql.com/products/administrator/>.

Ed. Copyright © Adrian Als & Charles Greenidge. 2003. University of West Indies at Cave Hill. 26 Mar. 2005
<http://scitec.uwichill.edu.bb/cmp/online/cs22l/prototype.htm>.

Webopedia. 4 Feb. 2005 <http://www.webopedia.com/TERM/R/RC4_RC5_RC6.html>.

07 Apr. 2005 <http://stylusinc.com/Common/Concerns/SoftwareDevtPhilosophy.php>.

Ed. James R. Chapman. February 2005. 06 Apr. 2005
<http://www.hyperthot.com/pm_sdm.htm>.

Ed. David B. Stewert. 1999. Embedded Systems Programming. 06 Apr. 2005
<http://www.embedded.com/1999/9910/9910feat1.htm>.

2005. Verizon. 19 Apr. 2005
<http://64.233.161.104/search?q=cache:lP4vbhSuLv0J:www2.verizon.net/help/dsl/%3Fcase%3D16526+msn+safe+credit+card+use+on+web&hl=en>.

Last Modified September 15, 2003. Ed. Adrian Als & Charles Greenidge. 2003. University of West Indies at Cave Hill. 26 Mar. 2005
<http://scitec.uwichill.edu.bb/cmp/online/cs22l/waterfall_model.htm>.

September 1999. 12 Apr. 2005 <http://www.website101.com/index.html>.

Feb. 2004. Whatis.com Target Search. 04 Apr. 2005
<http://whatis.techtarget.com/definition/0,,sid9_gci935732,00.html>.

Stylusinc. 07 Apr. 2005
<http://stylusinc.com/Common/Concerns/SoftwareDevtPhilosophy.php>.