Spring 2015

Project Plan

Version 1.0

CMIS 495

Andrew Bernier, Jason Foley, Brett Fry, John Livingston, Christopher Overby, Ina-Marie Sanabria, Charles Schultz

**University of Maryland University College**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 1/22/2014 | 1.0 |  | TeamX |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1 Project Summary 3

1.1 Purpose, Scope, and Objectives 3

1.2 Assumptions and Constraints 4

1.3 Project Deliverables 4

2. References 5

3. Definitions 5

4. Project Organization 5

4.1 External Interfaces 6

4.2 Internal Structure 6

4.3 Roles and Responsibilities 7

5. Managerial Process Plans 8

5.2 Work Plan 8

5.2.1 Work Activities 8

5.2.2 Schedule Allocation 9

5.3 Risk Mangement 12

6. Technical Process Plans 12

6.1 Process Model 12

# 1 Project Summary

The software development plan (SDP) provides an overview of the schedule, tasks, and resources required to create/develop and deliver the web store. This document will list in detail, the project assumptions, constraints, deliverables, and schedule. In summary it defines the technical and managerial processes to develop the web store and satisfy the product/project requirements.

## 

## 1.1 Purpose, Scope, and Objectives

The software system will be a web store designed to process user transactions and record sales for tax purposes. Per the statement of need/product requirements, the application/software will be used to allow users to make and process online payments, and to monitor expenses and profits. The web store will display products available for purchase. When a potential customer makes a purchase, it will have the ability to update, add, delete, and clear the cart. If the user wishes to check out, it will have the ability to enter the customer name, address, and phone number, card number, and process the payment. After check out it will store and record the user transaction in a database.

The objective of the WEB STORE is to increase sales and move a brick and mortar store online. By creating an online store and allowing customers to add, update, delete, and clear their carts, the application will increase sales because customers will no longer have to leave their homes to buy our products. By tracking the sales, it will increase the company’s ability to files taxes in a timely and more efficient manner. The added benefit is inventory control because items will be deleted from the inventory as they are sold, increasing the company’s ability to restock in a timely manner.

In order to accommodate both the needs of staff (employees and owners) and customers, the software should provide functionality specifically limited to the above areas. By not providing out-of-scope functionality, such as extended financial transactions, or information about other establishments, etc., the scope of the software will be specific to the needs of the customer/user.

The objectives of the WEB STORE are guided by efficiency, usability, and platform independence.

The deliverables for the completed product on delivery should include:

1. Software package consisting of:
   1. Product display
   2. Cart
   3. Financial transaction form for payment
   4. Ability to monitor expenses and profits
   5. Periodic checks on inventory (to warn if stock is low)
2. User guide/manual
3. Quick use pamphlet

Successful delivery of these features is dependent on user acceptance testing for milestones where the software is versioned (alpha, beta, version 1.0, etc.), successful automated testing to include destructive, functional, and unit testing of all units. Additionally, integration testing across the source code repository (revision control software/system), and guided walkthroughs of the user guide/manual and quick use pamphlet will be done and approved by the team before deemed acceptable or complete.

A list of product requirements will be available in the WEB STORE Software Requirements Specification, version 1.0.

## 1.2 Assumptions and Constraints

The development of the WEB STORE is designed around portability (platform independence). By focusing on the use of a platform independent object oriented programming language (Java), this software can be used on any operating system that supports Java (JVM) and can run on low power ARM or System on Chip (SoC) devices.

Assumptions for the WEB STORE are minimal due to the lean design process/model, and lack of reuse across the development cycle. Where appropriate, the development teams shall use free and open-source software (FOSS) to accelerate (or simplify) development, and make use of APIs and classes/libraries for specific functions (such as database connectivity and portability to other systems). In addition, the uses of the database or databases within the environment are entirely dependent on the IT infrastructure in use by the customer and can easily be changed based on the needs or desires of the customer. Testing for several relational database management systems (RDBMS) will be required to ensure compliance with these technologies and interoperability with the Java Persistence API (JPA), and will be referenced within the user guide/manual in the deliverables.

Interfaces with other software and required acquisitions of software are not applicable for the WEB STORE.

## 1.3 Project Deliverables

Project deliverables for the WEB STORE are focused on the following:

1. Software package consisting of:
   1. Product display
   2. Cart
   3. Financial transaction form for payment
   4. Ability to monitor expenses and profits
   5. Periodic checks on inventory (to warn if stock is low)
2. User guide/manual
3. Quick use pamphlet

The software delivery of the WEB STORE will consist of a Java application with the ability to run on the desired operating system(s) (Linux, Mac OS X, and Windows). This software will also contain example configuration files for database connectivity, showing configuration options for MySQL and PostgreSQL (the database used is irrelevant due to the use of JPA and the platform independence of Java, but minor changes might need to be made depending on implementation/infrastructure). The software and documentation will be made available electronically for download by customers.

# 2. References

The WEB STORE project management plan consists of the following documents and references:

1. IEEE Standard for Software Project Management Plans, standard 1058-1998
2. Software Development Plan, version 1.0 (this document) – TeamX
3. Software Requirements Specification, version 1.0 – TeamX
4. Software Design Document, version 1.0 – TeamX
5. Software Test Specification, version 1.0 – TeamX

# 3. Definitions

The following definitions and lexicon are referenced within the WEB STORE design and development plan:

* *Agile*: A lean development methodology used by the WEB STORE team to manage requirements on an iterative basis. The opposite of waterfall.
* *alpha testing*: Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
* *WEB STORE*: The Web Store Application
* *beta testing*: Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Beta versions are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has minimal faults or bugs.
* *Java Persistence API (JPA)*: Is a Java programming language application programming interface specification that describes the management of relational data in applications using Java Standard and Enterprise Editions.
* *Java Virtual Machine (JVM)*: An abstract computing machine.
* *relational database management system (RDBMS):* is a database management system (DBMS) that is based on the relational model.
* *System on Chip (SOC)*: An integrated circuit that integrates all components of a computer or other electronic system into a single device.
* *User experience (UX)*: involves a person's behaviors, attitudes, and emotions about using a particular product, system or service. User Experience includes the practical, experiential, affective, meaningful and valuable aspects of human–computer interaction and product ownership.
* *Waterfall*: a full project management and development methodology used to describe the process and workflow in a linear and fully resolved fashion. The opposite of agile.

# 4. Project Organization

This section specifies how the project team is organized for WEB STORE, for both external and internal interfaces. Overall, the project team is designed to be small in nature, efficient, and effective. The main focus of the projects organization is to streamline the development and communication process/structure between the team internally and validate project requirements.

## 4.1 External Interfaces

Due to the small and efficient design of the WEB STORE development team, external interfaces other than directly to the user (staff) and customer are not applicable (N/A).

## 4.2 Internal Structure

The WEB STORE development team structure will consist of a project lead, three software developers, a UX specialist, and two testers / technical writers. This team composition will create a structure allowing for direct development of the software requirements, design and modification of the user interface elements, and requirement and process management.

Figure 4.1: WEB STORE Development Team Structure and Process Flow

Developers

UX Specialist

WEB STORE Development Team

Program Manager, Technical Writers, and Customer Outreach Liaison

User Population (Requirements Pool)

As seen in figure 4.1, the WEB STORE development team is designed to encapsulate and isolate the process needs specific to each team member/group, while allowing for maximum flexibility and communication within the development team. In this diagram, the Developers and UX Specialist are highlighted in baby blue. The roles these members perform are directly responsible for the vital function of the software and interface construction. As a result, some shielding and isolation from the main process should be given to allow for maximum efficiency. This is done through communication directly to and from both the program manager for high level process and requirements management, as well as the customer outreach liaison to ensure the customer’s requirements are being created with the customer’s desires in mind. Additionally, this two-way flow ensures that communication between all elements of the team exists for optimum results and all team members are communicating with each other.

As mentioned above, isolation and separation is key to the efficiency of developers and designers. While this is important, it is also critical to ensure there is constant feedback to and from the customer to ensure the requirements are being met in a satisfactory and timely manner. To guarantee this process, both the program manager and customer outreach liaison will handle high-level customer engagement milestones. More important are the internal or low-level engagement milestones that drive the project along. These are conducted specifically by the customer outreach liaison directly with the customer on a regular schedule and information is passed back to the project manager and developers. This ensures that customer requirement questions and issues from the development team are handled directly and frequently, without having to stop developers from working, and providing a single point of contact with the customer, resulting in a streamlined communications process.

All software within the WEB STORE development cycle will make use of software version control, specifically implemented with Git using GitHub. Developers will ensure the following requirements are met for change management and version control to ensure quality assurance throughout the software development/programming process:

1. Source code is committed when any significant changes have been made or at least once every 12 hours, and more frequently if able.
2. Pushes to the primary repository must only occur after sufficient testing has been completed to verify that the changes work, and do not break builds.
3. Test infrastructure will make use of automated change management tools, specifically using Vagrant and Virtual Box. This will allow repeatable results for the creation of virtual environments that builds are deployed to, ability to test on multiple platforms, and remove team requirements for system administrators. It will also allow for the creation and restoration of snap shots of any given system or configuration.

To ensure that software has been tested and provide for a high-level of quality assurance, developers are responsible for providing complete unit, destructive, functional, and integration test suites for automated testing. These tests are to be built in JUnit to match the Java class design (See the Software Design Document (SDD) for additional details), and accurately match functionality or desired outcome at the method level during development. The repository commit process should also run automated tests which will force a 100% pass rate of code moving into the source code repository.

Where external interactions occur, such as database configurations, integration tests should be created to test the interoperability, functionality, and ensure that the software interacts as expected with the customer's RDBMS. For supported databases, each configuration should have an appropriate integration test that passes information correctly to and from the database and directly corresponds to the requirement and statement of configuration within the user’s guide.

Lastly, all automated test suites should match a corresponding requirement within the Software Test Specification (STS), and be committed and version controlled within the source code repository and monitored for success or failure of the commits.

## 4.3 Roles and Responsibilities

The WEB STORE development team is constructed of the following work roles and responsibilities:

One (1) Project Lead (Brett Fry): Performs the task of project oversight, requirements management, and scheduling of tasks/obligations for the team member(s). Will deliver project-related integrated master schedules (IMS) as well as conduct schedule conflict resolution to mitigate any risks associated with the waterfall methodology. Additionally, is also responsible for feedback from the customer base, interacting with the customer outreach liaison, resolving any major customer issues/disputes, and providing deliverables on-time at periodic milestones as dictated by the project schedule within the project lifecycle.

Three (3) Software Developers (Andrew Bernier, Jason Foley, and Charles Schultz): Performs software development tasks using the Java language. Must be fluent in the Java language, as well as libraries and methods used to implement software in a cross-platform process. Additionally, the understanding and configuration of data sources and connectivity to persist data in the WEB STORE will be required to validate and test requirements associated with saving data.

One (1) User Experience (UX) Specialist (Ina-Marie Sanabria): Performs the task of design, design efficiencies, and special internal/external requirements associated with the human/computer interaction (HCI) of the WEB STORE. This person shall provide mockups/diagrams based on user engagement and customer interaction/feedback to the development team for the user interface, as well as design and refactor designs related to all elements of the user interface. Additionally, must create/hold focus groups to determine the most efficient method and layout for accomplishing a task and pass the information back to the development team.

Two (2) Testers / Technical Writers (John Livingston, Christopher Overby): Will be responsible for providing timely and relevant feedback to the development team on the usability of software builds and relaying any issues or concerns as they occur. Additionally, the testers will be the primary authors of the user guide and other documentation.

# 5. Managerial Process Plans

This section of the Software Development plan highlights the work schedule and task structure required to meet the customer needs, requirements, and schedule as agreed to by both parties.

## 5.2 Work Plan

The following section discusses in detail the work breakdown schedule and project schedule for the WEB STORE development team.

### 5.2.1 Work Activities

The table on the following page constitutes the full breakdown of tasks under the master Work Breakdown Schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Task Name | SDLC Task Category | SDLC Sub-Task |
| 1 | Form Development Team | Analysis | Business Analysis |
| 2 | Form Project Plan / SRS | Analysis | Business Analysis |
| 3 | Form Initial User Guide | Design | Software Analysis |
| 4 | Develop / Deliver Test Plan | Design | Test Design |
| 5 | Develop / Deliver Project Design (SDD) | Design | Software Design |
| 6 | Develop / Deliver Phase 1 Documentation | Code | Interface / Service / Documentation Development |
| 7 | Develop / Deliver Phase 2 Documentation | Code | Interface / Service / Documentation Development |
| 8 | Develop / Deliver Phase 3 Documentation | Code | Interface / Service / Documentation Development |
| 9 | Final Bata Testing | Test | User Testing |
| 10 | End of Development & Testing | Test | User Testing |
| 11 | Complete User Guide(s) / Quick Use pamphlet | Code | Documentation Development |
| 12 | Deliver Product and Documentation | Code | Software Build |

Figure 5.2.1.1: Breakdown of Master Work Schedule

In the Master Work Schedule (Figure 5.2.1.1) above, the overall tasks are mapped to the calendar schedule, Figure 5.2.2.1, but are assigned by category in the software development lifecycle (SLDC) and their corresponding sub-task category. This corresponds to the full breakdown of work by task and subtask for the project lifecycle of the WEB STORE.

### 5.2.2 Schedule Allocation

In Figures 5.2.2.1 and 5.2.2.2, seen below, the entirety of the project schedule can be seen in a Gantt chart view (See attached GIF for more in-depth view). This view shows the dependency mapping between tasks, and isolates important milestones as called out by a diamond representation in the chart. Specifically, the entire schedule shows both project-related, development-related and delivery tasks as appropriate over the entire course of the project, which is approximately 8 weeks in duration. Due to the nature of the waterfall methodology in use (see section 6.1), changes in schedule to external forces, changes in requirements, or technical slips during the development cycle can lead to a range of time slips to the right, pushing the project past its agreed and intended delivery date of March.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Number | Task Name | Duration | Start | Finish |
| 1 | Form Development Team | 7 days | Mon 1/12/15 | Sun 1/18/15 |
| 2 | Form Project Plan / SRS | 7 Days | Mon 1/19/15 | Sun 1/25/15 |
| 3 | Form Initial User Guide | 3 Days | Mon 1/26/15 | Wed 1/28/15 |
| 4 | Develop / Deliver Test Plan | 5 Days | Wed 1/28/15 | Sun 2/1/15 |
| 5 | Develop / Deliver Project Design (SDD) | 7 Days | Mon 2/2/15 | Sun 2/8/15 |
| 6 | Develop / Deliver Phase 1 Documentation | 7 Days | Mon 2/9/15 | Sun 2/15/15 |
| 7 | Develop / Deliver Phase 2 Documentation | 7 Days | Mon 2/16/15 | Sun 2/22/15 |
| 8 | Develop / Deliver Phase 3 Documentation | 7 Days | Mon 2/23/15 | Sun 3/1/15 |
| 9 | Final Bata Testing | 3 Days | Mon 3/2/15 | Wed 3/4/15 |
| 10 | End of Development & Testing | 2 Days | Wed 3/4/15 | Thu 3/5/15 |
| 11 | Complete User Guide(s) / Quick Use pamphlet | 4 Days | Wed 3/4/15 | Sat 3/7/15 |
| 12 | Deliver Product and Documentation | 1 Day | Sun 3/8/15 | Sun 3/8/15 |

Figure 5.2.2.1: Project Schedule

Figure 5.2.2.2: Gantt chart for Project Schedule

## 5.3 Risk Mangement

In order to accommodate for and mitigate the risk of database configuration and implementation mismatch, the development team must be able to provide the following:

1. Specific documentation stating what database software is supported by the WEB STORE (this shouldn’t matter much since JPA will be used).
2. Generate and apply specific tests against supported RDMBS.
3. Provide configuration examples for the supported RDBMS software.
4. Work with the customer, and provide support, for RDBMS software that is out-of-scope for integration.
5. Generate scenarios in documentation that explain possible configurations for the database(s). An example of this would be the use of SQL, which is similar in implementation to MySQL.

By enforcing a rigid set of examples and tests against the supported RDBMS software, working with the customer to ensure that these match specific requirements for development, and providing helpful support when issues arise, much of this risk is mitigated, providing a specific path on how to compensate for the risk.

# 6. Technical Process Plans

This section addresses the process and mechanics, by which the WEB STORE development team will plan for, perform analysis of, construct, and test the source code based on the customer’s requirements.

## 6.1 Process Model

The methodology to be used by the WEB STORE development team shall be constructed in a waterfall fashion for the purpose of program management. Internally, the development team may organize and develop within the assigned periods for deliverables in a pseudo-agile fashion, building epics, stories and tasks in conjunction with the overall planning strategy for the team. This model is heavily influenced by the Department of Defense model for government acquisition planning with lean development phases. This model helps to achieve large scale planning for acquisition and management, but provides more flexibility to the development teams tasked with implementation.

The process in place accommodates for a start activity based on project assessment. This provides for the development of both sub-processes and requirements which are used to generate milestones over the course of the project, and to assign responsibility of development tasks throughout the team. The conclusion of the project, also listed as a milestone, serves as the final delivery of the project after multiple cycles of refinement, refactoring, and user acceptance testing.

6.2 Methods, Tools, and Techniques

The WEB STORE development team, following the waterfall methodology for program management in section 6.1, which will also require a standard process by which development, user feedback, testing, integration and documentation is performed. The following standards are to be implemented and adhered to throughout the lifecycle of the project:

1. Language: Java, associated classes/libraries, and APIs
2. Database Implementations: MySQL and PostgreSQL
3. Source Code Repository and Version control: Git, GitHub
4. Documentation: Adobe Acrobat, LibreOffice, and Microsoft Word.
5. Testing: Junit, and TestNG
6. Build and Deploy: Maven, and Glassfish

All Java source code is to be written using standard Java or Google Java coding style and conventions, the database languages will use the standard formatting and coding conventions based on the language specification, and committed only upon successful completion of all relevant unit, function, destructive, and integration tests. Any builds not conforming to this standard are to be rolled back to the prior commit/version then fixed, verified with automated testing tools, and pushed forward to the repository. The commit will also include additional information specifying what was causing the error, why it was written/coded the way it was, what was done to correct the error, and the actions to be taken in the future to prevent it from occurring again.

Each build associated with a milestone in the project schedule will be built using automated build processes in Maven and Glassfish and supplied to the testers for discussion of user expectations to include usability. Then to provide a demonstration of functionality to ensure the desired requirements have been or are being met, and user acceptance testing is being done. Failure to provide a timely build for the testers will result in an internal review by all parties to address any issues leading to the issue.