**Project Planning for Willy’s Widget Company’s Inventory Management System**

**REVISION HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No. | Revision Description | Modified  By | Modification Date | Comments |
| 1. | Initial draft created | SungEun Kim | 2014/05/23 |  |
| 2 | Teamwork | Tommy Park | 2014/05/26 |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1. Project Scope 5](#_Toc388936213)

[1.1 Project Name 5](#_Toc388936214)

[1.2 Project Goal & Strategy 5](#_Toc388936215)

[1.3 Duration 5](#_Toc388936216)

[1.4 Success Criteria 5](#_Toc388936217)

[2. Development Process 6](#_Toc388936218)

[2.1 Agile Process 6](#_Toc388936219)

[6.2 Activity Definition 7](#_Toc388936220)

[6.3 Role Definition 8](#_Toc388936221)

[3. WBS 9](#_Toc388936222)

[4. Schedule 11](#_Toc388936223)

[5. Roles and Responsibility 12](#_Toc388936224)

[6. Time Log 13](#_Toc388936225)

[6.1 Time Log Category 13](#_Toc388936226)

[6.2 Time Log vs. WBS Work Item 13](#_Toc388936227)

[7. Test Plan 14](#_Toc388936228)

# Project Scope

## 1.1 Project Name

Willy’s Widget Company’s Inventory Management System

## 1.2 Project Goal & Strategy

The project goal is development of Willy’s Widget Company’s inventory management system based on a robot-assisting semi-automated warehouse system. This system should process integrated customer orders which would be placed in a laptop.

Major values could be added by this project are

* Fast delivery time in customer’s perspective
* Correct and easy management of warehouse inventories in manager’s perspective
* Lowering worker’s load in worker’s perspective
* Lowering labor cost in stockholder’s perspective

Although above all values can be major goal of our project, we will basically focus on the correct and easy management of warehouse inventories because system’s feature extension should be based on correctness. Moreover, we have only 5 weeks for developing system. Of course, we will review if this assumption is correct in each iteration splint.

In addition, customer order system will be limited to laptop in this project, but we will design the system to be extended for mobile environment. The reason of this is that mobile handset usage is increasing more and more. This can increase system extensibility.

## 1.3 Duration

- Draft Plan and ADS in Korea: May 12th ~ May 26th

- Product Analysis/Design/Implementation: May 27th ~ Jun 27th

## 1.4 Success Criteria

- To develop & deliver reliable and extensible WWC’s IMS on time (Jun 27th).

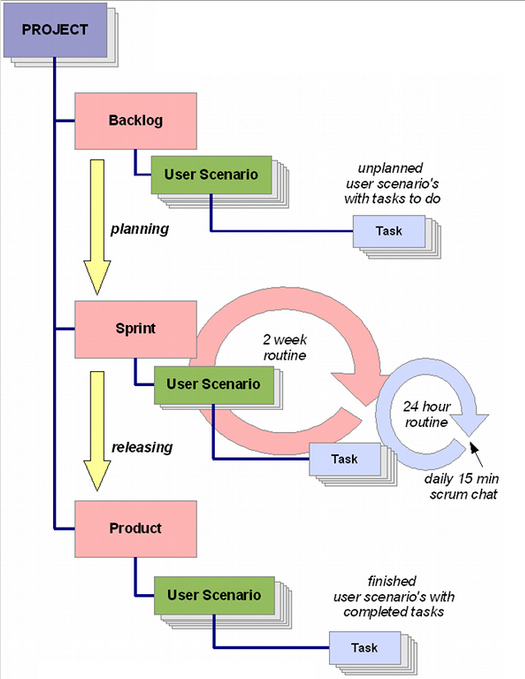
- To apply the practices learned from the lectures to this project.

# Development Process

We develop using Agile Method to allow maximum flexibility and control.

2.1 Agile Process

Agile takes a flexible approach to development. A high priority is “to satisfy the user through early and continuous delivery of valuable software.”



2.2 Activity Definition

| **Name** | **Description** |
| --- | --- |
| Project | * Embrace all activities the team to deliver goal/deliverable. |
| Scenario | * Compact descriptions of the expected behavior of the application. * Deduced from Use Cases * All user scenarios together describe the complete way the application working |
| Backlog | * Contains all user scenarios that has not yet planned |
| Task | * Piece of work that one of the team members needs to fulfill to accommodate a user scenario. * Each task belongs to only one user scenario. * Task larger than Scrum term will be spilt in a smaller task as the goal will be complete them within Sprint * Task can contain: * Description * Links to discussions, file attachments * Link to a demo/ test environment * Optional link to a bug tracking system * The planned amount of hours to realize * Actual number of hours it took to finish |
| Sprint | * A fixed period of time, usually 2-4 weeks, in which the team performs tasks belonging to a user scenario. * When all sprints are released, the project is complete and the application is ready. |
| Sprint Planning Meeting | * The planning is the meeting before the start of each sprint. * Team selects a set of user scenarios which the highest priority that the team should be able to implement within the fixed amount of time the sprint will take. * This is the process of moving user scenarios from the backlog to the sprint. |
| Scrum Meeting | * Scrum meeting is a daily 15mintes chat, usually at 9:00. * Only the Scrum master and Developers participate in this meeting: * What’s done? * What will do? * Any issues? |

2.3 Role Definition

| **Name** | **Description** |
| --- | --- |
| Product Owner | * Represents the voice of the customer and is accountable for ensuring that the Team delivers value to the business. * Write customer-centric items (typically user stories), prioritizes them, and adds them to product backlog. * Scrum teams should have one Product Owner and while they may also be a member of the Development Team, it is recommended that this role not be combined with that of Scrum master * Write user stories * Prioritize user stories * Accepting user stories after concluding a iteration |
| Scrum master | * Scrum master is not a team leader but acts as a buffer between the team and any distracting influences. * Ensures that the Scrum process is used as intended. * Question the Project Owner to ensure they fully understand the requirements of the Product owner and end-user * Host the iteration meetings * Help the product owner participate defects and even make sure there is time reserved to fix bugs during the iteration. |
| Team | * Responsible for delivering the product * Made up of people with cross-functional skills who do the actual work: * Analysis * Design * Develop * Tests * Document |
| Client Stakeholder | * Contribute with User Scenarios. * One or more persons from the client will be involved in reviewing each completed iteration |

# WBS

We can break down work based on User Scenario and each component.

| **Component** | **Task ID** | **Tasks** | **Scenario** | **Estimation** |
| --- | --- | --- | --- | --- |
| Customer Application | WTC-01 | display the Type of the Widgets | UC 01 | 15 |
| WTC-02 | send the order to the Warehouse | UC 01 | 15 |
| WTC-03 | display the status of order | UC 01 | 30 |
| WTC-04 | can handle error cases | UC 01 | 30 |
| Supervisor Application | WTS-01 | display the Order status | UC 02 | 15 |
| WTS-02 | provide an interface to handle backordered order | UC 02 | 30 |
| WTS-03 | provide an interface to create new inventory entry for Inventory Stations | UC 03 | 30 |
| WTS-04 | The Supervisor Application shall provide an interface to add the number of widgets in Inventory Stations. | UC 03 | 30 |
| WTS-05 | display the robot information | UC 04 | 15 |
| WTS-06 | When an error happens, the supervisor can order recovery action. | UC 05 | 15 |
| WTS-07 | display the Inventory information | UC 07 | 15 |
| Warehouse System | WTW-01 | shall take the orders from the customers | UC 01 | 55 |
| WTW-02 | maintain customer orders with Queue (FIFO) | UC 05 | 15 |
| WTW-03 | Maintain backordered orders with another Queue. | UC 05 | 15 |
| WTW-04 | move the robot to the next station till all the items of the order are loaded | UC 05 | 50 |
| WTW-04 | the warehouse system debits corresponding widgets from entries in Inventory | UC 05 | 15 |
| WTW-05 | move the robot to the shipping station after the order is filled | UC 05 | 30 |
| WTW-06 | shipping worker presses the button, the warehouse system updates order status and robot status | UC 05 | 30 |
| Total |  |  |  | 450 |

# Schedule

By following Agile Process we can split project two Sprints with 2 weeks periods.

During the first Sprint, we will design and implement interfaces of core functionalities of placing an order. At second Sprint, we will implements the manager features of the System.

At the final periods, we will prepare demo scenario based on Sprint results.

| **Period** | **Task** | **Deliverable** |
| --- | --- | --- |
| 5/12~ 5/25 | Project Scoping | Project Plan  Backlog |
| 5/26 ~ 6/6 | **Sprint #1**  Core-Functionalities | * Warehouse System which can handle order by interacting with warehouse H/W system * Customer Application which can place an order to warehouse * Supervisor Application which can add inventory items |
| 6/6 | **Sprint#1 Demonstration** | |
| 6/9 ~ 6/20 | **Sprint #2**  Monitor Features | * Warehouse system can update warehouse status * Supervisor Application which can monitor order, robot, inventory status |
| 6/20 | **Sprint#2 Demonstration** | |
| 6/23 ~ 6/27 | Product | Warehouse System, Customer Application, Supervisor Application,  User Guide, Supervisor Guide |
| 6/27 | **Product Demonstration** | |

# Roles and Responsibility

| **Role** | **Assignment** | |
| --- | --- | --- |
| Product Owner | All members | |
| Scrum master | Sungeun Kim | |
| Team | Customer Application | Kyung Lang Park |
| Supervisor Application | Jinyoung Du  Vijay Rachabattuni |
| Warehouse System | Byungseon Shin  Jugwan Eom |
| Client Stakeholders | Philip Bianco  Antony Lattanze | |

# Time Log

## 6.1 Time Log Category

Time log category is as below :

* Planning
* Requirements
* Design
* Implementation
* Testing

## 6.2 Time Log vs. WBS Work Item

Each work item of WBS will has one attribute corresponding to time log category. So, we update and accumulate time log by WBS excel sheet. Finally, accumulated time log will be compared to estimated time in planning phase.

# Test Plan

Test will follow iterative agile process also. Each test case list will be described, implemented and executed incrementally.

* 1st week
  + Unit Test Case List Description(Basic)
* 2nd week
  + Unit Test Case List Description/Implementation(More Detail)
  + Unit Test Execution
  + Integration Test Case List Description/Implementation (Basic)
* 3rd week
  + Unit Test Execution
  + Integration Test Case List Description/Implementation (More Detail)
  + Integration Test Execution
  + System Test Case List Description/Implementation (Basic)
* 4th week
  + Unit Test Execution
  + Integration Test Execution
  + System Test Case List Description/Implementation (More Detail)
  + System Test Execution
* 5th week
  + Unit Test Execution
  + Integration Test Execution
  + System Test Execution