COMARCH
SYSTEMY INFORMATYCZNE

Pocket XL

Project Plan

Implementation prototype of new retail sales module of CDN XL

DOCUMENT DESCRIPTION

Prepared for:	IIT – IPRO 349
Short name:	PP
Version:	1.0.5
Last modification:	2011-02-23
Authors:	Mirosław Jedynak, Francisco Garcia, Sebastian Babel, Tomasz Kijasm Michał Waśniowski
Company:	COMARCH SA
Page number:	11
Document status:	For internal use IIT and Comarch

CHANGE HISTORY

Version	Date	Change's author	Chapter	Change range
1.1	6/25/07	Francisco	All	Changed/deleted headings and filled in information
1.2	6/26/07	Sebastian	All	Read through and checked grammar
1.3	6/26/07	Francisco	All	Fixed paragraph formatting
1.4	6/26/07	Mirosław Jedynak	All	Fixed minor mistakes
1.5	6/27/07	Francisco	1	Rearranged objectives; added info to background

TABLE OF CONTENTS

I. Project definition

I.1. Project objectives

The objective of this group is to produce barcode scanning software for warehouse acceptance. This means that when a warehouse receives an item (or items) information is generated about that item from its barcode that lets a warehouse worker know where the item should be stored in the warehouse. At the end of this project we should have a solution that is ready for verification of specifications by a Comarch deployers.

Our software should meet several criteria:

- Integrated into CDN XL which is Comarch ERP solution software.
- Make use of its algorithm to improve the efficiency of a warehouse by directing the employees to where items should be stored in the warehouse.
- Foolproof to minimize the effects of human error.
- Capable of multinational use and will include a language pack for Polish and English.
- Have a minimal learning curve in order to be an effective and marketable product.

Some aspects of the software should be customizable after the prototype stage such as the format of barcode it will read, what type of mobile device it will run on and the appearance of the graphical user interface. The customization capabilities will not be implemented in this project since our software will be the initial release.

I.2. Project background

The software to be developed in this project is intended for use by warehouse employees to improve the efficiency of the warehouse item storage logistics. Our team is not involved in the process of contacting customers who will eventually be using the product. That task is left to Comarch's deployers who will verify that all required specifications have been fulfilled.

This product is intended as an improvement on current modules of warehouse acceptance. Current modules require excessive manual input of information that is time consuming. They also do not provide information on the proper storage location for items.

Some warehouse solutions currently available from companies such as Zebra Technologies and TEKLYNX offer specialized barcode printing used for tracking. The problem with this approach is the added cost of purchasing the barcode printing equipment and software. Since nearly all products already carry a printed barcode on the packaging or on the item itself, it would be more useful to use

those barcodes to assist with proper warehouse storage. Our software will do just that. It can potentially remove the cost of printing equipment that would require eventual maintenance and purchasing of print materials.

In order to develop our product we will require several software programs. We will be using a bar code generator program in order to test the bar code reader. The reader (a Windows Mobile Device) will use a wireless network connection to communicate with a server to retrieve information about items. To utilize the wireless network we will use a web service as a communication interface. The program is basically an add-on to CDN XL which uses application program interface (API).

The possible cost to the customer would be in the time it would take for warehouse employees to learn to use the bar code scanning software. Another cost would be in the physical installation of the program into the warehouse system.

I.3. Project risks

One risk arises from the combination of an inexperienced team given a short amount of time. We are not familiar with C# programming which is the language we will be using to implement the project. Assigning this project to an inexperienced team runs the risk of not being completed in the six weeks allotted. This would be due to the time it will take for the team members responsible for the majority of the programming to become familiar with C#. There is a low-medium probability of non-completion of the project due to inexperience. The outcome would be to present as much work as was completed at the end of the term.

The method of implementation we are using involves our work being reviewed by one of Comarch's deployers at the end of every week for meeting certain requirements. Our tasks the following week depend on feedback from the deployer. Not getting feedback at the beginning of each week would delay our ability to set new tasks for the rest of the week and continue progress of the project. For several unforeseen reason there is a high probability that this would happen at least once during this project. This would be cause for us to make up for the lost time.

Given that we are relying on computers and software to implement this project we are always in the shadow of possible hardware or software failure that would delay our progress. It is difficult to predict the extent of damage to the project a hardware or software failure would have.

I.4. Project expectations

The project that we are working on is due in 6 weeks. The real development is expected to last for 5 weeks, because the 6th week is only for minor changes and final presentation. During these 5 weeks we are going to produce a working prototype each week. On the beginning of the 6th week, we want to have a working application that matches the primary objectives. This application is going to be a prototype of a new CDN XL module. This module is a solution for warehouses. It should help

warehouse operators in dealing with deliveries and package distribution in warehouse. The application is indented to run on small, portable devices with network connection and barcode scanning possibilities.

Our application is intended for a wide range of clientele, and specific user information beyond project requirements is unavailable at this time. The final project may be adjusted and deployed in a particular company.

II. Organization of the project

II.1. Functions and responsibilities in project

II.1.1. Project Manager

Mirosław Jedynak – oversee entire project by communicating with the deployer and assigning tasks to interns. Assist the interns with their tasks as needed.

II.1.2. Mobile Device Programming

Sebastian Babel – design and implement GUI of mobile device.

II.1.3. Server & mobile device communication

Michał Waśniowski – design and implement communication between the mobile device and the server.

II.1.4. Server program

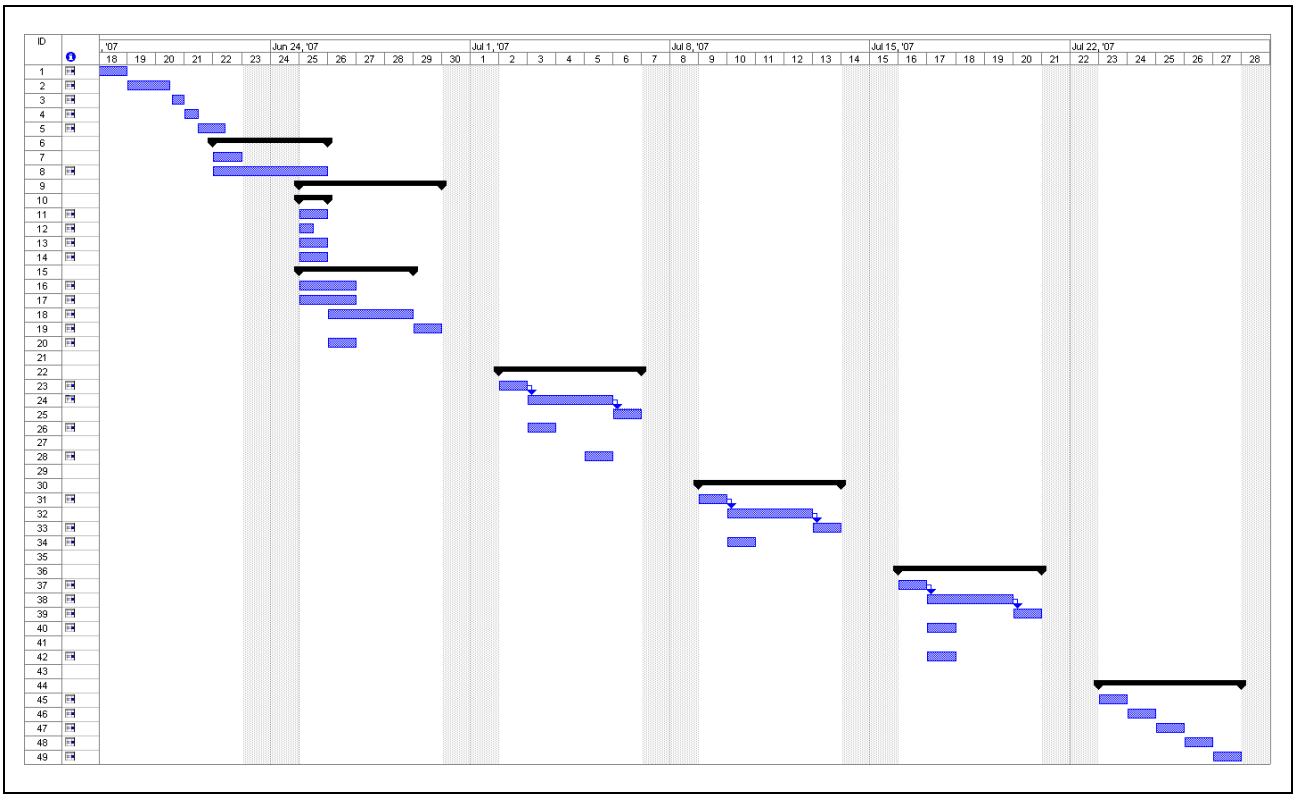
Tomasz Kijas – implement location algorithm on the server.

II.1.5. Testing and verification

Francisco Garcia – run test cases on program at the end of each stage and document any errors.

III. Project schedule

Base project schedule is presented below:



Detailed analysis stage is agreed as follows:

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Comarch Orientation	1 day	Mon 6/18/07	Mon 6/18/07		
2	Software Install / Computer Setup	1.5 days	Tue 6/19/07	Wed 6/20/07		
3	ERP Slideshow / Visual Studio tutorial	0.5 days	Wed 6/20/07	Wed 6/20/07		
4	CDN XL Setup	0.5 days	Thu 6/21/07	Thu 6/21/07		
5	Official Project Briefing	0.88 days	Thu 6/21/07	Fri 6/22/07		
6	**Use Case / Project Plan Formation**	**2 days**	**Fri 6/22/07**	**Mon 6/25/07**		
7	Use Case	1 day	Fri 6/22/07	Fri 6/22/07		Sebastian Babel,Michal Wasniowsl
8	Project Plan Assembly	2 days	Fri 6/22/07	Mon 6/25/07		Cisco Garcia
9	**Week 1**	**5 days**	**Mon 6/25/07**	**Fri 6/29/07**		
10	**Analysis**	**1 day**	**Mon 6/25/07**	**Mon 6/25/07**		
11	Use Case adjustment	1 day	Mon 6/25/07	Mon 6/25/07		Sebastian Babel
12	Testing Case Requirements	0.5 days	Mon 6/25/07	Mon 6/25/07		Cisco Garcia,Sebastian Babel
13	Risk Analysis	1 day	Mon 6/25/07	Mon 6/25/07		Cisco Garcia
14	Web Services/C# learning	1 day	Mon 6/25/07	Mon 6/25/07		Michal Wasniowski,Tomasz Kijas
15	**Production**	**4 days**	**Mon 6/25/07**	**Thu 6/28/07**		
16	GUI Design	2 days	Mon 6/25/07	Tue 6/26/07		Sebastian Babel
17	High Level Architecture Sketch	2 days	Mon 6/25/07	Tue 6/26/07		Michal Wasniowski,Tomasz Kijas
18	Coding	3 days	Tue 6/26/07	Thu 6/28/07		
19	Integration	1 day	Fri 6/29/07	Fri 6/29/07		
20	Conference Call 15.45	1 day	Tue 6/26/07	Tue 6/26/07		
21						
22	**Week 2**	**5 days**	**Mon 7/2/07**	**Fri 7/6/07**		
23	Analysis	1 day	Mon 7/2/07	Mon 7/2/07		
24	Production	3 days	Tue 7/3/07	Thu 7/5/07	23	
25	Integration	1 day	Fri 7/6/07	Fri 7/6/07	24	
26	Conference Call 15.45	1 day	Tue 7/3/07	Tue 7/3/07		
27						
28	Midterm Review	1 day	Thu 7/5/07	Thu 7/5/07		
29						
30	**Week 3**	**5 days**	**Mon 7/9/07**	**Fri 7/13/07**		
31	Analysis	1 day	Mon 7/9/07	Mon 7/9/07		
32	Production	3 days	Tue 7/10/07	Thu 7/12/07	31	
33	Integration	1 day	Fri 7/13/07	Fri 7/13/07	32	
34	Conference Call 15.45	1 day	Tue 7/10/07	Tue 7/10/07		
35						
36	**Week 4**	**5 days**	**Mon 7/16/07**	**Fri 7/20/07**		
37	Analysis	1 day	Mon 7/16/07	Mon 7/16/07		
38	Production	3 days	Tue 7/17/07	Thu 7/19/07	37	
39	Integration	1 day	Fri 7/20/07	Fri 7/20/07	38	
40	Conference Call 15.45	1 day	Tue 7/17/07	Tue 7/17/07		
41						
42	IPRO Day Tips/Guidelines Session	1 day	Tue 7/17/07	Tue 7/17/07		
43						
44	**Final Week**	**5 days**	**Mon 7/23/07**	**Fri 7/27/07**		
45	Poster/Exhibit	1 day	Mon 7/23/07	Mon 7/23/07		
46	Abstract/Brochure	1 day	Tue 7/24/07	Tue 7/24/07		
47	Presentation Creation	1 day	Wed 7/25/07	Wed 7/25/07		
48	Final Report	1 day	Thu 7/26/07	Thu 7/26/07		
49	IPRO Day	1 day	Fri 7/27/07	Fri 7/27/07		

Specific taks for each week are not available as they will be set after receiving feedback from the deployer at the beginning of each week.

IV. Methodology and procedures during project implementation

- **Typical spiral model**

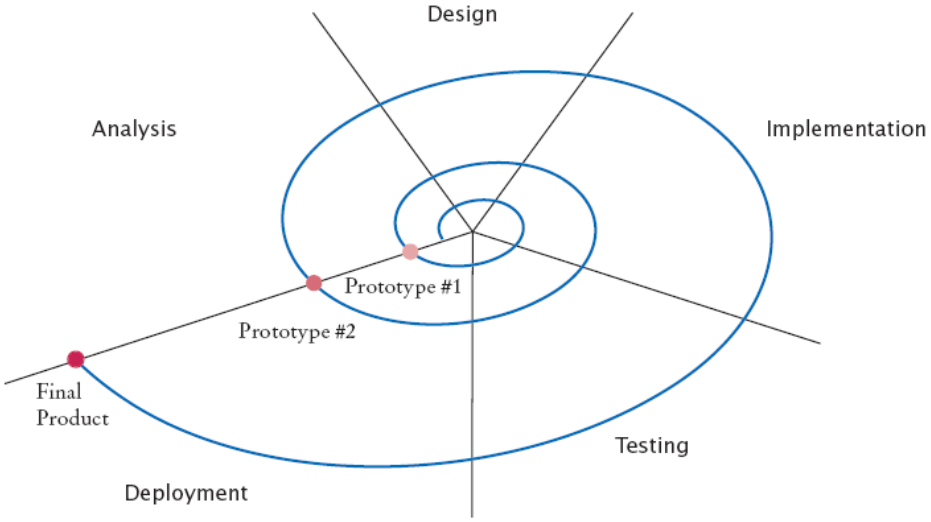


Figure 2 A Spiral Model

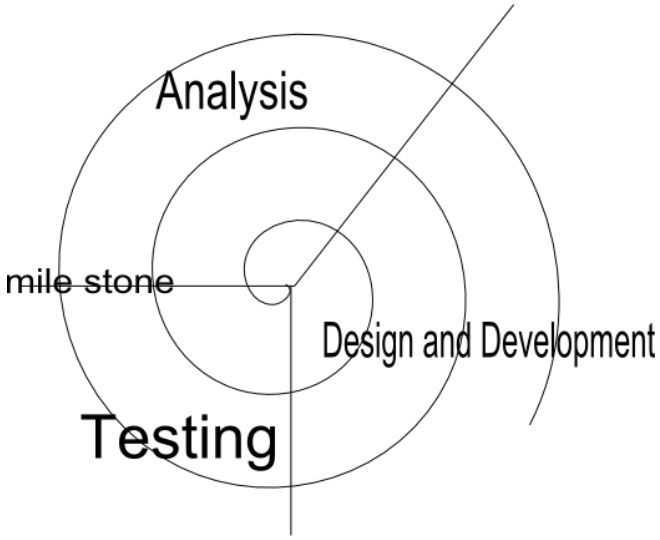
Spiral model general definition

The spiral model, also known as the spiral lifecycle model, is a systems development method (SDM) used in information technology. This model of development combines the features of the prototyping model and the waterfall model.

Characteristic features of typical spiral model

- Breaks development process down into multiple phases
- Early phases focus on the construction of prototypes
- Lessons learned from development of one prototype can be applied to the next iteration
- Problem: can lead to many iterations, and process can take too long to complete

Spiral model used by our team. It's adapted to our needs.



Description of phases in our model

- Analysis - gathering requierments
- Design and Development - architecture and solutions used during iteration
- Testing - verification of requirements from analysis phase and comments on functionality, what has impact on analysis in next iteration

Agile – short description

Agile is conceptual framework for undertaking software engineering projects that embraces and promotes evolutionary change throughout the entire life-cycle of the project.
It is one of the most modern methodologies. It was created in the mid 1990s by people who didn't like "heavyweight" methods like waterfall model of development. They found waterfall model slow, demeaning and inefficient.

Project requirements in Agile

In Agile methodology programmers doesn't know exactly their requirements. They change them when customer tests and gives feedback about products. Depending on this feedback the project group makes a new analysis and produces another iteration. At the end of each iteration the team should produce working software. That's the main reason why we chose the Agile methodology. Each Friday we are supposed to have a working prototype ready.

IV.1. Communication in the project

Agile methods emphasize face-to-face communication, over written documents. Team works usually are in one place and can easily talk with each other. Our team is working in one room and we

sit next to each other, thus verbal communication is easy among us. We will also utilize email accounts provided by Comarch to receive tasks and for other communication relevant to the project.

IV.2. Documentation in the project

In Agile working software is the most important part. Teams make, relative to other methodologies, very little documentations and comment code a lot. In Agile it is important to make clear code.

In order to track progress we will be completeing IPRO deliverables (in addition to this project plan) such as a midterm and final reports, a final presentation, and meeting minutes. We will also utilize feedback from the deployer each week.

V. Products tests and verification in project

The deployer will send our team the initial requirements and the initial implementation tasks will be assigned and completed. At the end of each week our team will produce working code and will run initial test cases to verify that the program performs as intended. This version of the program will then be presented to the deployer. The deployer will provide feedback on the finished code and based on that feedback the team will return to the previous iteration of the project and make any necessary corrections or will move on to the next iteration.

VI. References

http://www.langdonsystems.com/default.asp
http://www.zebra.com/id/zebra/na/en/index.html
http://www.sun.com/software/solutions/rfid/
http://www.teklynx.com/

VII. List of Appendix

Appendix 1 – Project plan in MS Project format