### 

Assignment number: Assignment 01

Assignment unique number: ***332555***

Student number: 5516-073-5

Last name: Mtombeni

Initials: P.M

Name of Course: Research Methods and Proposal

Course Code: HPCOS81

Due date: 30 June 2014

Word Count: 5000

Lecture Name**:** Jan Mentz

Introduction..............................................................................................................p4

Part I…….................................................................................................................p4

1 Overview Summary..............................................................................................p4

1.1 Project Summay .............................................................................................p4

1.1 .1 Purpose,scope and Objectives...............................................................p4

1.1 .2 Assumptions and Constraints.................................................................p5

1.1 .3 Project Deliverables ...............................................................................p5

1.1.4 Schedule and budget summary..............................................................p5

1.2 Evolution of the Software Management Plan ...................................................p6

2 Reference Materials..............................................................................................p6

3 Definitions and Acronyms ....................................................................................p6

4 Project organization...............................................................................................p7

4.1 External interfaces..........................................................................................p7

4.2 Internal interfaces...........................................................................................p7

4.3 Roles and responsibilities ..............................................................................p8

5 Managerial process plans .....................................................................................p8

5.1 Start-up plan ...................................................................................................p8

5.1.1 Estimation Plan ..................................................................... .................p8

5.1.2 Staffing Plan ………………......................................................................p9

5.1.3 Project Staff and Training Plan ................................................................p9

5.1.4 Project Staff and Training Plan ................................................................p9

5. 2    Work Plan .....................................................................................................p9

5.2.1 Work Activities.................................................................................................p9

5.2.2 Schedule allocation.........................................................................................p10

5.2.3 Resource Allocation ........................................................................................p10

5.2.4 Budget allocation.............................................................................................p10

5.3.1 Requirements Control Plan ............................................................................p10

5.3.2 Schedule Control Plan.....................................................................................p10

5.3.3 Budget control plan..........................................................................................p11

5.3.4 Quality control plan..........................................................................................p11

5.3.5 Reporting Plan ................................................................................................p11

5.3.6 Metrics Collection Plan ...................................................................................p11

5.4 Risk Management Plan .....................................................................................p11

5.5 Project close-up plan ……..……………..............................................................p12

6 Technical processes.............................................................................................p12

6.1 Process Model...................................................................................................p12

* 1. Methods, Tools and Techniques ...........................................................p13
  2. Infrastructure Plan .................................................................................p13
  3. Product Acceptance Plan ......................................................................p13

7 Supporting plan................................................................................................….p13

7.1 Configuration management plan......................................................……………p13

7.2Test plan..............................................................................................................p13

7.3 Documentation Plan...........................................................................................p14

7.4 Quality Assurance Plan .....................................................................................p14

7.5 Reviews and Audits plan....................................................................................p14

7.6 Problem resolution plan......................................................................................p14

7.7 Subcontractor Management plan.......................................................................p14

* 1. Process Improvement Plan...........................................................................p14

8.1 Addition plans.....................................................................................................p14

8.2 Available Hardware and Software Resources ..................................................,p15

8.3Project Dependencies and Closure Requirements ..............................................p15

Use Case Diagram........................................................................……………....p16 Use-case Description: Oil Distribution Corporation………..........................................p17

Look up Item availability details……...........................................................................p17 *Create new Order* ………………….............................................................................p17

Update Order……………….........................................................................................p18

Order return notice………............................................................................................p19

Maintain Customer Account Information…..................................................................p19

Catalog request……………………………..…...............................................................p19

Promotion package details …….................................................................................p20

Customer charge adjustment …………………….........................................................,p20

Catalog update details………………………………… ..................................................p20

Special promotion details………………………………….........................……………....p21 New catalogue details…………………………………...……..........................................p21

Order Fulfilment notice………..…...............................................................................p22 Back Order Notice…………..........................................................................................p22

Produce transaction summary report............................................................................p22

Produce order summary report.....................................................................................p23

Use Case Realisation(Order Summary)……….............................................................p24

Use Case Realisation (Update order)…..**…..….**............................................................p25

Use Case Realisation (Transaction Summary)…..............................................p26

Use Case Realisation (New Order)…..……………......................................................,p27

Use Case Realisation(New Order is successful)..**……** ................................................p28

Use Case Realisation(Maintain Account Information).......................................p29

Glossary ………………………………………………………….......................................p30 PART II ………………………………………………………….........................................p32 Full bibliographic details….…………………………………….......................................p32

Area of study……………….……………………………………......................................p32

Relates to information..………………………………….......................................p33 PART III…………………..………………………………….......................................p33

The purpose of the Software Engineering Code of Ethics………...............................p33

The importance of ethical behaviour ………………………………….………………......p34

The adoption issues of the Software Engineering Code of Ethics……….…………......p34 IEEE and ACM……………………………………………………………...……………......p34

Conclusion…..……………………………………………………………...….…………......p36Reference…………...……………………………………………………...….…………......p37

**Introduction**

This assignment focusses on characteristics of Software Project management Plan. The important aspects or concept of Software Project management Plan is discussed in this assignment are for example Project Summary, Budget and Test Plan. This paper will tackle software engineering methodologies, processes, and frameworks .This assignment also focusses on relationship between software design patterns and software reusability. It also covers the benefits and software engineering challenges associated with Software Engineering Code of Ethics

### Question1

### 1 Overview

### 1.1 Project summary

### 1.1.1 Purpose, Scope and Objectives

The purpose of this project is to design and implement a software product, which will assist the CIO of a regional oil distribution corporation to carry out all accounting functions of the company to provide online information to the head office staff regarding orders and inventory in various company storage tanks. Second goal, which will be incorporated during the project, is to produce informative, structured documentation of the product itself, the software creation and design processes, as well as the project management process.

Requirements for the course will fulfilled when the customer accepts the results and deliverables.

The software product in question is not directly related to any other named software or projects and will act as an independent product.

The requirements of the delivered software product are specified in detail in the Software

Requirements Specification document.

### 1.1.2. Assumptions and Constraints

In this project a technique called is used to bring a new perspective into analyzing process data. The software will be able to detect various trends in time perspective and thus help the person conclude what just happened or did not happen in the process.

The deadline for the software product is set to 31 July 2014. Keeping that in mind, the scheduling will be quite tight, but also realistic. Estimated amount of work time is about four weeks (480 hours). The software product will be one of the first of its kind and therefore everything has to be done within the project few if any pre-made software components will be used, except for the standard libraries and interfaces which are described in detail in the technical process plans section.

The client has proven to have a significant amount of dedication towards this project. That said, there should not be any risks involving client's premature project cutbacks

## 1.1.3. Project deliverables

Below a list of items that have to be delivered to the client is given. Everything will be delivered on paper form except for source code and executables that will only be delivered electronically.

The items that will be delivered to the client are:

* User Requirements Document
* Software Requirements Document
* Prototype
* Detailed Design Document
* Software User Manual
* Software Transfer Document
* Source code and executable programs
* Software Project Management Plan
* A copy of the entire project repository on compact disk
* UML Diagrams
* A path that user can access the system
* SPMP documentation
* System Requirement Analysis Software System Proposal
* Software Design Document
* User Documentation Manual,
* Specification
* The work product(s) to be delivered to the client. The quantities, delivery dates, and delivery locations are specified in the project agreement.

The complete product, including the above, will be delivered 160 working hours after the project commences.

## 1.1.4. Scheduling and Budget Summary

The hours worked per week (R40 per hour) are 35 and are scheduled as follows:

Requirements workflow (8 hours, 4 team members, R59404.65)

Analysis workflow (40 hours, 4 team members, R60000.00)

Design workflow (40 hours, 4 team members, R60000.00)

Testing workflow (40 hours, 4 team members, R30494.10)

Implementation workflow (40 hours, 4 team members, R60000.00)

Post Maintenance workflow (40 hours, 4 team members, R29898.75)

The total development time is 40 hours and total cost is **R300000.00 .**

## 1.2. Evolution of the Software Project Management Plan

Although initially written alongside the Software Requirements Specification document this Software Project Management Plan will evolve as the project goes on.

Any major change in this document has to be ratified by the client. All changes should be documented to keep the project management plan correct and up to date.

## Reference Materials

SRS

[SRS] Software Requirements Specification, Project

SDS

[SDS] Software Design Specification

DTW

[DTW] Berndt, D.J. & Clifford, J. 1996. Finding patterns in time series: A dynamic programming approach. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy R. (eds.) Advances in Knowledge Discovery and Data Mining. Menlo Park, AAAI Press / The MIT Press. P. 229-248.

REVIEW

[REVIEW] Research Scientist, Computer Laboratory, University of South Africa.

## Definitions and Acronyms

|  |  |
| --- | --- |
| UR | User Requirements Phase |
| CSC | Computer Software Component |
| SRS | Software Requirements Specification |
| SPMP | Software Project Management Plan |
| SDS | Software Design Specification |
| ICD | Interface Control Document |
| JDK | Java Development Kit, Sun Microsystems, Inc. |
| JVM | Java Virtual Machine |
| JRE | Java Run-time Environment |
| JNI | Java Native Interface |
| Swing | A collection of inter-architecture windowing components |
|  | for Java |
| STL | Standard Template Library for C++ |
| MoM | Minutes of meeting |
|  |  |
|  |  |
| HTML | HyperText Markup Language |
| UML | Unified Modeling Language |

## 4. Project Organization

## 4.1. External Interfaces

No external organizations will take part in this project, other than in the roles of client and project team.All the work on this project will be performed by Mandla and the team. Mandla will meet weekly with the client to report progress and discuss possible changes and modifications.

## 4.2. Internal Structure

|  |
| --- |
|  |
| **Figure 2:** Project Groups |

The organizing model in this particular project will be somewhat flat. The number of people participating is so small that this is the only way to go.

Project lead group, consisting of Mr.Mandla and team from University of South Africa will review all work accomplished by the project staff and uses a democratic team approach.

Each member of the project staff will have their own responsibilities according to their prior knowledge of the subject, but since fast and effective communicating is not a problem between the staff members, consulting each other will not become an issue.

Each member of the assigned project staff will be responsible for all documentation concerning the work modules they have to accomplish. Consulting other group members is an essential part of the documentation and design processes.

All project members will be communicating with each other, in order to introduce a flat organization model. Each person will have to acknowledge their own skills so that others may acquire information from them, whenever needed. And vice versa, one will not have to hesitate to ask questions from their peers. Component-Based Development mandates that project members and component users give detailed feedback to others of their work, otherwise good results are not to be expected.

## 4.3. Roles and Responsibilities

Rough division of roles between the staff members:

* Mandla project management for: wrapper module and interface design   
  software requirements specification
* Mandla project management for: DLL design and implementation  
  wrapper module implementation
* Other team members project management for: GUI design and implementation software design specification

### 5. Managerial process plans

### 5.1 Start-up plan

### 5.1.1. Estimation Plan

However, scheduling constraints dictate that the staff needed in this project are expected to deliver at time. Requirements and technical design, implementation and testing phases will take a maximum of a few weeks each, shortest being only three days. A big emphasis on documentation demands some time from implementation, but that will be considered in scheduling.

The overall working time spent by the core group in this project is expected to be four man-months, that being near the artificial milestone of 160 hours. Amount of time spent on each task is described in section 3.2.1

### 5.1.2. Staffing Plan

### As long as the project will be commencing, all the team members will be involved.

Specify the numbers and types of personnel required to conduct the project.

Name Function E-mail

Mr P.M. MTOMBENI 2000032207 2000032207@student.uovs.ac.za

Mr Team member 1 2002097273 2002097273@student.uovs.ac.za

Mr Team member 2 2002091649 [2002091649@student.uovs.ac.za](mailto:2002091649@student.uovs.ac.za)

Function not yet being specified

**5.1.3. Resource Acquisition plan**

All necessary software package of the product will be handed in on the disk

* + 1. **Project Staff and Training Plan**

No extra staff will be hired during this project. Three members of the core group will be enough for a project of this scale.

Project staff will not be trained especially for or during this project, but naturally some new information concerning for example the algorithm and the user interface has to be acquired in order to deliver a usable product.

Purpose of this project is to also learn and develop each participant's skills in software projects' task fields.

**5. 2    Work Plan**

Specify the work plan, dependency relationships, resource requirements, allocation of budget and resources to work packages, and a project schedule. Much of the content may be in appendices that are living documents, updated as the work proceeds.

Work activities need to be broken down to the level of being able to identify necessary resources, estimate duration; list products produced acceptance criteria and predecessor and successor work activities.

### Work Activities

Complete SPMP

Complete Use Case Scenarios

Complete Formal Specification Document

Implement System in Alloy

Verify System

Prepare Presentation

* + 1. **Schedule allocation**

Leaving aside delivery deadlines, work from different activities can proceed

in parallel. After an initial planning, design, and feasibility check process, it

is planned to progress development, testing and documentation on different

areas of functionality in turn, directed by milestones.

* + 1. **Resource Allocation**

Specify the allocation of budget and resources to the various project functions, activities, and tasks.

There are no significant resource allocations for this project apart from stated

time.

**5.2.4 Budget allocation**

Predictions for the minor financial costs of this project are stated in section **5.3.3**

5.3.1 **Requirements Control Plan**

Formal and non-formal reviews of the code and documentation will be held both inside the project team and the client, in order to make sure that everything required will actually be done well and in time. There is no formal requirements control plan. Development is directed towards the requirements specified in the proposal UML document, but can be varied as the problem is better understood.

**5.3.2 Schedule Control Plan**

Project staff will be in contact daily to bring delays or fallbacks immediately to others' knowledge. Code milestones will occur at the end of each module's coding phase, to keep formal track records and ease risk assessment and evaluation. Weekly meetings among the core group will also be held. Since the project deadlines are very firm, milestone scope will be adjusted towards a self-consistent and stable software build before each delivery.

**5.3.3 Budget control plan**

NOT APPLICABLE 990.0775

Requirements 40 hours @ 1485.12/man hour R59404.65

Specification/ Analysis 40 hours @ 1500/man hour R60000.00

Design 40 hours @ 1500/man hour R60000.00

Implementation 40 hours @ 1500/man hour R60000.00

Other 40 hours @ 1509.82/man hour R60392.85

---------------------- -------------- --------------

**Total 160 hours R300000.00**

**5.3.4 Quality control plan**

Team members are encouraged to identify errors in their codes and reaching the clients requirements.

### 5.3.5 Reporting Plan

A memo of each formal review will be provided alongside with the project's formal documentation. A final report of the whole project will also be written.

Each member will report any disturbances in a subtask to the project manager. This report should contain at least the time spent so far and an estimate of the work still needed. After a subtask has ended, the project manager will also receive a formal notification with information of the time spent.

### 5.3.6 Metrics Collection Plan

Emphasizing on metrics collection in this project would result delays and distortion in the creative process. Working time statistics and scheduling accuracy will be collected. Testing results will provide formal metrics.

## 5.4 Risk Management Plan

Most important risk factors in this specific project are:

1. Difficulties in implementation
2. Overoptimistic scheduling
3. Wrong direction of emphasis
4. Personal issues (due to the small team)
5. Inexperienced project management

Factor 1 is the most likely risk to realize. In order to keep this from happening, there are a few guidelines for the project members to keep in mind:

* Be honest to yourself and to others
* Be critical about your skills, do not excess them
* Ask questions

If the first risk realizes at any point, the resolution is to take advantage of any person who has knowledge in the problematic area. This includes the use of Usenet newsgroups or other bulletin board systems found on the Internet. If, and only if the problem is a real show-stopper and development of the software depends on its solution, means of project's management level resolution plans will take place.

Overoptimistic scheduling (2) and inexperienced management (5) together form the second most dangerous risk cluster. Avoiding risk realization will be much easier if management completes the tasks "by the book" and does not cut to own conclusions at every turn.

Two risk factors (3,4), that are left, are the most difficult to minimize, because the direction of emphasis will show up only at the end of the project and personal issues are almost never predictable.

If any of the most likely risks, namely overoptimistic scheduling, seem to realize at a fast pace, there are at least three options to choose from:

1. end the project at that stage,
2. leave the project at hold until all the members of project team are available, or
3. continue under-staffed until the project is finished

of which the last one is most preferred resolution plan both from the client's and the project staff's point of view.

**5.5 Project close-up plan**

**5.5.1** **Close-up Plan**

Normally, all documents and code would be provided to your advisor in electronic form to allow for maintenance. The other items will be part of the final Studio Exposition. State that fact here.

1. **Technical processes**

## 6.1 Process Model

Unified Process will be used refer to UML documents.

This subsection of the SPMP shall determine the relationships among major project functions and activities by specifying the timing of major milestones, baselines, reviews, work products, project deliverables, and sign-off that span the project. The process model or system development life cycle (SDLC) approach may be described using a combination of graphical and textual notations. Software Project Management Plan. This is the controlling document for managing a software project, and it defines the technical and managerial processes necessary to deliver the project requirements. The process model may be described using a combination of graphical and textual .Each

milestone should be reached with a consistent and useful set of functionality,

test cases for implemented functions, and reasonable confidence in code coverage

The process model must include project initiation and project termination activities. Development will follow an incremental, milestone-driven method.

## 6.2 Methods, Tools and Techniques

For the Software Design Specification the software will be modeled using Microsoft Office Visio to draw UML diagrams.

Two main components of the software product will be programmed in different languages: The DLL will be implemented in C#. Code for the graphical user interface will be written in C#.

C# will be used as the Swing form design program.

## 6.3 Infrastructure Plan

No hardware investments will be made for this project. The software development will be done using existing workstations and network infrastructure.

## 6.4 Product Acceptance Plan

High quality of documentation is essential for product acceptance and coursework grading.

Before acceptance, the software MUST fulfill the steps of unified process

**7 Supporting plan**

**7.1 Configuration management plan**

**Not applicable**

**7.2 Test plan**

**Not applicable**

## 7.3 Documentation Plan

Documents to be written during this project are the project plan (this document), software requirements specification, technical specification, the results and the final report.

Each document will be written , using standard packages to standardize formatting. An HTML conversion of the finished documents will be provided for the client with the help of HTML.

No separate user's manual will be written, because the target audience will be either familiar with the algorithm at hand, or learning it.

## 7.4 Quality Assurance Plan

Quality of the software product will be verified and improved during the development process using techniques such as formal reviews and peer inspection. Formal reviews will be held internally in the group when a task is about to end or has just ended.

Each delivered document will be formally reviewed with the client

**7.5 Reviews and Audits plan**

A Software Quality Assurance Plan will be developed following recommended departmental standards. See section 1.1.2.

.1 for its delivery date.

Problem Resolution plan

**7.6 Problem resolution plan**

A problem resolution system will be determined closer to project release. If

the project is publicly adopted then problem reporting and resolution will

be managed through the systems.

**7.7 Subcontractor Management plan**

NOT APPLICABLE

## 7.8 Process Improvement Plan

Project management documentation and a development journal will be maintained through the project, and examined to identify excessive rework or

opportunities for better process.

**8.1 Addition plans**

Security –A username and password will be needed to use the product

Training-Training will be performed by development team at the time of delivery .Because the product is straightforward to use ,1 day should be sufficient for training .Development team will answer questions at no cost for the first year of use

Maintenance-Corrective maintenance will be performed by the team at no cost for a period of 12 month. A separate contract will be drawn up regarding enhancement.

### 8.2 Available Hardware and Software Resources

Hardware and software resources are well-suited for a project of this scale. However, the client can loan a laptop computer for development and testing with needed software for short periods of time. Development material repository will reside on a computer at the computer engineering laboratory in the University of South Africa.

## 8.3 Project Dependencies and Closure Requirements

The project is a standalone project: it is not dependent on any other projects. By July 30th 2014, when all the software and documentation deliverables have been finished, reviewed and accepted, the project will be finished

**State Chart**

**Use Case Diagram**



# Use-case Description: Oil Distribution Corporation

## Look up Item availability details

|  |
| --- |
| **Brief Description** The Look up Item availability details use case enables the customer to check whether item is available or not |
| **Step-by-Step Description**   1. Customer call in to check item and restock date by giving the following details:  * Name of item * Size of item      1. Phone–order representative ask him/her if s/he has ordered by phone before 2. Phone–order representative ask customers give their details  * Customer ID number * Customer Name  1. Phone–order representative ask customers what catalog are they looking at  * Product ID  1. Phone–order representative informs the customer if the item is available or not   **Note**: Details of the algorithm for determining the availability will be obtained later. |

## Create new Order

|  |
| --- |
| **Brief Description**  The create new order use case allows the customer to order an item that he wants and needs |
| **Step-by-Step Description**   1. Customer creates new order  * Via web * Via phone  1. Phone/mail-order representative enters the following details in an order screen  * Customer ID * Date of order * Date of arrival * Product ID for the order  1. The system should look up the following  * Customer details to validate the ID number but it should also display the name and address to the representative so the representative is confident the number is correct.  1. The system displays the description of the product  * Product ID * Product name * Product category * Product size * Product colour  1. As the representative's attention shifts back and forth from the mail-order form to   the computer screen, he or she compares the name and product description from  the system with the information on the form to confirm that everything is correct.  **Note**: Details of the algorithm for determining the new order will be obtained later. |

**Update Order**

|  |
| --- |
| **Brief Description**  The update order use caseallows the customer to change or cancel order |
| **Step-by-Step Description**   1. **C**ustomer may decide to    * change existing order or    * cancel the existing order 2. Customer calls phone-order representative 3. Customer tells the phone-order representative that he wants to    * Change existing order or    * Cancel the existing order 4. Phone-order representative may    * changes order or    * cancels the existing order on the system made by the customer 5. Change confirmation is sent to back to the customer   **Note**: Details of the algorithm for determining the new order will be obtained later. |

**Order Status details**

|  |
| --- |
| **Brief Description**  Order Status details use caseallows the customer and CIO to check order status |
| **Step-by-Step Description**   1. Customer calls the Phone-order representative 2. Phone-order representative check:    * shipment on the computer and    * back order status on the computer 3. Phone-order representative tells the customer about the status of the order whether it is changed/cancelled or it is in the process of shipping   **Note**: Details of the algorithm for determining the order status details will be obtained later. |

**Order return notice**

|  |
| --- |
| **Brief Description**  Order return notice use case allows the customer to return item |
| **Step-by-Step Description**   1. Customer changes his mind 2. Customer returns order to inventory 3. Phone-order representative:    * receives returned order    * updates customer information   **Note**: Details of the algorithm for determining the order return details will be obtained later. |

**Maintain Customer Account Information**

|  |
| --- |
| **Brief Description**  The maintain customer account information use case enables the customer to update account information |
| **Step-by-Step Description**   1. Customer updates account information which may includes    * Changed address(Home/Work)    * Marital status    * Phone number    * Bank details 2. Customer informs Phone-order representative about updated account information |

**Catalogue request**

|  |
| --- |
| **Brief Description**  Catalog request use case allows prospective customer to request catalog |
| **Step-by-Step Description**   1. Prospective customer to request catalog by giving the following details    * Customer ID    * Date of order    * Date of arrival 2. Phone-order representative prepares catalog 3. Phone-order representative sends to catalog prospective customer using  * E-mail * Post * Telephone |

**Promotion package details**

|  |
| --- |
| **Brief Description**  Promotion package details use case allows Market to send Promotional materials to customers |
| **Step-by-Step Description**   1. Market prints address information which includes    * Physical address    * Residential address    * Postal address 2. Market track promotions 3. Market representative sends Promotional materials to customers via  * E-mail * Post * Phone |

**Customer charge adjustment**

|  |
| --- |
| **Brief Description**  Customer charge adjustment use case enables Manager to adjust customer charges |
| **Step-by-Step Description**   1. Phone-order representative makes corrections to customer accounts  * Fault customer ID * Fault customer name * Fault customer address * Fault customer phone number  1. Phone-order representative sends corrections of customer accounts to manager 2. Manager adjust customer charges    * Correct customer ID    * Correct customer name    * Correct customer address    * Correct customer phone number |

**Catalog update details**

|  |
| --- |
| **Brief Description**  Catalog update details use case allows Merchandising to update catalog |
| **Step-by-Step Description**   1. Phone-order representative changes seasonal items ,offerings and description by:    * Add    * Delete and    * Change prices 2. Phone-order representative sends changed catalog to merchandising 3. Merchandising updates catalog details |

**Special promotion details**

|  |
| --- |
| **Brief Description**  Special promotion details use case enables Merchandising to create special product promotion |
| **Step-by-Step Description**   1. Phone-order representative track contents and mailings of promotions via:    * Web    * Phone 2. Phone-order representative sends special product promotions to Merchandising 3. Merchandising creates special product promotions |

**New catalogue details**

|  |
| --- |
| **Brief Description**  New catalog details use case allows Merchandising to create new catalog |
| **Step-by-Step Description**   1. Merchandising use computer generated catalog to prepare catalog active report which contains:    * Customer ID    * Address    * Product ID    * Phone number 2. Merchandising creates new catalog details |

**Order Fulfilment notice**

|  |
| --- |
| **Brief Description**  Order Fulfilment notice use case enables shipping to fulfil order |
| **Step-by-Step Description**   1. Phone-order representative updates order by entering:    * Customer ID    * Address    * Product ID    * Phone number 2. Phone-order representative prints label which contains    * Customer ID    * Address    * Product ID    * Phone number 3. Phone-order representative sends notice or order change details to shipper 4. Shipping fulfils order |

**Back Order Notice**

|  |
| --- |
| **Brief Description**  Back Order Notice use case allows shipping to identify Back Order |
| **Step-by-Step Description**   1. Shipping uses back order notice for out-of-stock items    * Via Web    * Via Post 2. Shipping sends back order notice to the phone-order representative 3. Phone-order representative receives back order notice |

**Produce transaction summary report**

|  |
| --- |
| **Brief Description**  The produce transaction summary report use case enables CIO and head office staff to obtain information summarising previous orders that occurred or to detect new trends in the oil market. |
| **Step-by-Step Description**   1. The following report must be generated on demand(question marks in the name of the transaction, or date of order must be printed  * Transaction report   The system displays all transactions that occurred during the past year.  The output in this order:  Date of transaction  Name of order  Classification of transaction  This report is sorted by classification and by date of transaction. |

**Produce order summary report**

|  |
| --- |
| **Brief Description**  The produce order summary report use case enables CIO and head office staff to obtain information summarising previous orders that occurred or to detect new trends in the oil market. |
| **Step-by-Step Description**   1. The following report must be generated on demand(question marks in the name of the order, or date of order must be printed  * Order report   The system displays all orders that occurred during the past year.  The output in this order:  Date of order  Name of order  Classification of order  This report is sorted by classification and by date of order. |

Use Case Realisation(Order Summary)



Use Case Realisation (Update order)



Use Case Realisation (Transaction Summary)

Use Case Realisation (New Order)



Use Case Realisation(New Order is successful)



Use Case Realisation(Maintain Account Information)

**Glossary**

**application architecture plan**

a description of the integrated information systems that the organization needs to carry out its business functions

**automation boundary**

the separation between the automated part of a system and the manual part of a system

**business process reengineering**

a technique that seeks to alter the nature of the work done in a business function, with the objective of radically improving performance

**communication support systems**

support systems that allow employees to communicate with each other and with customers and suppliers

**customer relationship management (CRM)**

processes that support marketing, sales, and service operations involving direct and indirect customer interaction

**decision support systems**

**(DSS)**

support systems that allow a user to explore the impact of available options or decisions

**enterprise resource planning (ERP)**

a process in which an organization commits to using an integrated set of software packages for key information systems

**executive information systems (EIS)**

information systems for executives to use for monitoring the competitive environment and for strategic planning

**functional decomposition**

dividing a system into components based on subsystems that in turn are further divided into subsystems

**information system**

a collection of interrelated components that collect, process, store, and provide as output the information needed to complete business tasks

**information systems strategic plan**

the plan defining the technology and applications that the information systems function needs to support the organization’s strategic plan

**Architectural design**– Broad design of the overall system structure; also called general design or conceptual design.

**Business logic layer**

**Centralized architecture**– Architecture that locates all computing resources in a central location.

**Client**

**Clustered architecture (328)** – A group of computers of the same type that share processing load and act as a single large computer system.

**Computer network**– A set of transmission lines, equipment, and communication protocols to permit sharing of information and resources.

**Data layer**

**Detail design**– Low level design that includes the design of specific program details.

**Distributed architecture**– Architecture that deploys computing resources in multiple locations connected by a computer network.

**Extranet**– an intranet that has been extended outside the organization to facilitate the flow of information.

**Interface designers**– Specialists in user-interface design; also called usability consultants or human factors engineers.

**Internet**– A global collection of networks that use the same networking protocol—TCP/IP.

**Intranet**– A private network that is accessible to a limited number of users, but which uses the same TCP/IP protocol as the Internet.

**Local area network (LAN)**– A computer network in which the distances are local, such as within the same building.

**Middleware**

**Multicomputer architecture**– A group of dissimilar computers that share processing load through specialization of function.

**Multitier architecture**– Architecture that distributes application-related software or processing load across multiple computer systems.

**Network diagram**

**Server Single-computer architecture** – Architecture that employs a single computer system executing all application-related software.

**Wide area network (WAN)** – A computer network spread across large distances, such as a city, state, or nation.

**World Wide Web (WWW), or Web**– A collection of resources such as files and programs that can be accessed over the Internet using standard protocols.

**PART II**

a) Provide full bibliographic details relating to the article in HARVARD notation

Bernhard Bauer, J¨org P. M¨uller, and James Odell. Agent UML: A formalism for specifying

multiagent software systems. In P. Ciancarini and M. Wooldridge, editors, *Agent-Oriented*

*Software Engineering — Proceedings of the First International Workshop (AOSE-2000)*.

b)Identify the area of study. (The introduction of the article will often help you to do this.) For example: The success of a software development project depends on capturing stakeholders’ needs in a specification.

**Methodology means a way of developing a software product .** The success of a software development project depends on methDOology(UML) and conceptual design.Modeling has been a cornerstone in many traditional software development methodologies for decades. The use of object-oriented modeling in analysis and design started to become popular in the late 1980s, producing a large number of different languages and approaches. Lately, UML has taken a leading position in this area, partly through the standardization of the language within the Object Management Group (OMG).We give an assessment of UML (version 1.4), highlighting both the positive aspects and the areas where improvement is needed.

The development of software requires a highly iterative design methodology which balances and synchronizes multiple design disciplines in order to provide an optimum solution which best meets the needs and requirements of the customer. While different design methodologies may vary in detail, most begin with a conceptual design phase which is focused on understanding the problem, weighting the customer requirements, generating feasible alternatives, and comparing alternatives in order to make major conceptual design decisions. Once a concept is selected, the preliminary design phase begins. During this phase, the designers continually refine their focus to conduct more detailed analysis in order to refine design parameters across multiple disciplines.

Throughout the evolution of system design, great advancements have been made in developing analysis tools to assist designers during the preliminary and detailed design phases. However, very few tools exist to assist designers during the conceptual design phase. Most analysis tools are very discipline or concept specific, and many are far too cumbersome to use to compare vastly different design concepts in a timely manner. Consequently, many conceptual decisions must be made qualitatively. Information from the preliminary design loops are fed back into the conceptual loop making the process iterative. In this way, conceptual decisions may be ratified or rejected from information gathered during preliminary design. However, a poor conceptual design

1 .decision that is not discovered until the completion of a preliminary design iteration will become very expensive to change and may jeopardize the critical path timeline of the project. This accentuates the need to make accurate conceptual design designs.

**c) State how this article relates to information in the prescribed book and the prescribed articles. 2 x 5**

**PART III**

**1.The purpose of the Software Engineering Code of Ethics.**

The Software Engineering Code of Ethics and Professional Practice, intended as a standard for teaching and practicing software engineering, documents the ethical and professional obligations of software engineers. The code should instruct practitioners about the standards society expects them to meet, about what their peers strive for, and about what to expect of one another. In addition, the code should inform the public about the responsibilities that are important to the profession. Adopted by the Computer Society and the ACM—two leading international computing societies—the code of ethics is intended as a guide for members of the evolving software engineering profession. The code was developed by a multinational task force with additional input from other professionals from industry, government posts, military installations, and educational professions.

**1.Purpose of the Code**

1. Is intended as a standard for teaching and practicing

software engineering,

2. Documents the ethical and professional obligations of

software engineers.

3. Instructs practitioners about:

• The standards society expects them to meet,

• About what their peers strive for

• About what to expect of one another.

4. Informs the public about the responsibilities that are

important to the profession.

**1. The benefits associated with Software Engineering Code of Ethics**

**1. Attracts new employees**

– Attracts dedicated and committed employees that want to be involved in an organization that produces quality software.

**2. Promotes a good image for a company**

– This let the public know that the company works for the public good and proudly accepts this responsibility.

**3. Promotes a Professional Image**

– Increases public respect your company and improve the quality of software it produces.

**4. Increases Public Confidence**

• Informs the public that their interests are met in full confidentiality and with high standards.

**5. Promotes the Deployment and adoption at the**

**international level**

• A code adopted across the planet by all software engineers and their employers

• Software engineers will make these principles a practice of each day and produce quality products.

**2. The importance of ethical behaviour during the maintenance phase of software development.**

Because of their roles in developing software systems, software engineers have significant opportunities:

– To do good or cause harm,

– To enable others to do good or cause harm,

– To influence others to do good or cause harm

Software products are developed and maintained by humans.if those individuals are hard working ,intelligent,sensible, up to date,and above are ethical ,then the chances are good that the way the software products they develop and maintain will be satisfactory. Unfortunately, the converse is equally true.

1. **The adoption issues of the Software Engineering Code of Ethics**

**The software engineering challenges associated with Software Engineering Code of Ethics**

**Ethical dilemmas**

Disagreement in principle with the policies of senior management.

Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.

Participation in the development of military weapons systems or nuclear systems.

**Ethical Issues**

Ethics and the IS Professions

Ethical Issues Related to Computers and Information Systems

Codes of Conduct

The Rules

1. Growing importance of software in society

2. Increasing public awareness

3. Software is an invisible and intangible threats

4. Profitability, safety and security

5. Increased awareness of legislators (i.e. Governments)

6. 'Formalization’ of a software engineering profession

1. The current developments in the effort to make software engineering into a profession(hint: initiative started jointly by IEEE and ACM in 1993)

Most societies for profesionals have a code of ethics to which all its members must adhere. The two major societies for computer professionals ,the Association for C omputing Machnery(ACM) and the Computer Society of the Institute of Electrical and Electronics Engineers(IEEE-CS) jointly approved a Software Engineering Code of Ethics and Professional Practice as the standard for teaching and practicing software engineeringIEEE/CM,1999].It is lengthy ,so short version,consisting of preamble and eight principle ,was also produced .This is the short version

Software Engineering Code of Ethics and Professional Practice 2(Version 5.2)

As recommended by the IEEE-CS/ACM Joint Task Force on

Software Engineering Ethics and Professional Practices

Short Version

Preamble

The short version of the code summarizes aspirations at a high level of abstraction;the clauses that are included in the full version give examples and details of how these aspirations change ,the details can become legalistic and tedious ;without the details, the aspirations can become high sounding but empty; together ,the aspirations and the details form a cohesion code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession.In accordance with their commitment to the health, safety and welfare of the public,software engineers shall adhere to the following:

1.Public- Software engineers shall act consistently with the public interest

2Client and Employer-Software engineers shall act in a manner that is in the best interest of their client and employer consistent with the public interest.

3.Product –Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4.Judgement-software engineers shall maintain integrity and independence in their professional judgement

5. Management –Software engineers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6.Professional- Software engineer shall advance the integrity and reputation of the profession consistent with the public interest.

7Colleagues –Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice pf the profession.

The code of ethics of other societies for computer professional express similar sentiments .It is vital for the future of our profession that we adhere rigorously to such codes of ethics.

8 SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

**Conclusion**

This assignment focussed on characteristics of Software Project management Plan. The important aspects or concept of Software Project management Plan was discussed in this assignment are for example Project Summary, Budget and Test Plan. This paper will tackled software engineering methodologies, processes, and frameworks .This assignment also focussed on relationship between software design patterns and software reusability. It also covers the benefits and software engineering challenges associated with Software Engineering Code of Ethics

**Bibliography**

M. Wooldridge, N. R. Jennings, and D. Kinny. A methodology for agent-oriented analysis

and design. In *Proceedings of the Third International Conference on Autonomous Agents*

*(Agents 99)*, pages 69–76, Seattle, WA, May 1999.

Ives, B., Hamilton, S., and Davis, G. B., “A Framework for Research in Computer-Based Management

Information Systems,” ***Management Sciences,*** Vol. ***26,*** No.9, September 1980, pp. 910-934.

Juan, T., Pearce, A. and Sterling, L.,(2002). Extending the Gaia Methodology for Complex Open

Systems, Proceedings of the First International Joint Conference on Autonomous Agents

and Multi-Agent Systems (AAMAS), p3-10, Bologna, Italy.

Golfarelli, M., Rizzi, S. (1998). A methodological framework for data warehouse design.

In: Proc. DOLAP, pp. 3–9.

Anderson, R.E., Johnson, D., Gotterbarn, D., Perrolle J.,(1993). ‘Using the New ACM

Code of Ethics in Decision Making’, Communications of the ACM.

Fuxman, A. Pistore, M. Mylopoulos, J. and Traverso, P. ( 2001). ‘‘Model checking early requirements specification

in Tropos,’’ in Proceedings of the 5th IEEE International Symposium on Requirements Engineering,

Toronto, CA.

Gordon, V.S., Bieman, J.M. (1995). Rapid prototyping: Lessons learned. IEEE Software

12(1), 85–95.

Basili, V. R.. Rombach, H. D.(1988). The TAME Project: Towards Improvement-Oriented Software Environments. IEEE Trans. on Software Engineering, 14(10):758-773.

- Gotterbarn, D., Miller, K., Rogerson, S. (1999). ‘Computer Society and ACM

Approve Software Engineering Code of Ethics’, IEEE Computer.

Gotterbarn, Miller, and Rogerson. (1997) “Software Engineering Code of Ethics” , (November /Vol. 40, No. 11, Communication of the ACM)

Wooldridge, .M. and Jennings, N. R.( 1995). Intelligent agents: Theory and practice. *The Knowledge*

*Engineering Review*, 10(2):115–152,

Estefan´ıa Argente, Ghassan Beydoun, Rub´en Fuentes-Fern´andez, Brian

Henderson-Sellers, and Graham Low. (2011)Modelling with agents. In Marie-

Pierre Gleizes and Jorge Gomez-Sanz, editors, *Agent-Oriented Software*

88 *Engineering X*, volume 6038 of *Lecture Notes in Computer Science*, pages 157–168. Springer Berlin / Heidelberg.

Giorgini,P. Mylopoulos , J. Giunchiglia , F. and Bresciani ,P. (2001 ).‘‘Agent-oriented software

development: A case study,’’ in S. Sen, J. P. Mu¨ ller, E. Andre and C. Frassen, (eds.), Proceedings of the

Thirteenth International Conference on Software Engineering – Knowledge Engineering (SEKE01),

Buenos Aires, ARGENTINA, June 13–15,

Giunchiglia,F. Mylopoulos, J. and Perini A. ‘‘The Tropos.( (2002).Software development methodology:

Processes, models and diagrams,’’, Springer- Verlag, in press.

Gotterbarn, Miller, and Rogerson”.(1999).Communication of the ACM) .The adoptions issues of the Software Engineering Code of Ethics. The Code Adoption Published(Vol. 42, No. 10,)

Gotterbarn, D., Miller, K., Rogerson, S. (1997), “Software Engineering Code of Ethics”, Communications of the ACM 40, 11 pp. 110-118.

Gotterbarn, D., Miller, K., Rogerson, S.(1999) “Software Engineering Code of Ethics is Approved”, Communications of the ACM 42, 11 pp. 102-107.

Gottenbar, D. (Jan. 1999).“A Positive Step Toward a Profession: The Software Engineering Code of Ethics and Professional Practice”, AMC SIGSOFT Software Engineering Notes 24, 1 pp. 9-14.

Perini, .A. Bresciani, P. Giunchiglia, F. Giorgini P., and Mylopoulos, J.( 2001). ‘‘A knowledge level software

engineering methodology for agent oriented programming,’’ in Proceedings of the 5th Int. Conference

on Autonomous Agents, Montreal CA, ACM.

Perini, A., Bresciani, P., Giunchiglia, F., Giorgini, P and Mylopoulos, J., A knowledge

Level.(2001). Software engineering methodology for agent oriented programming. Proceedings of

Autonomous Agents, Montreal CA