Cyber Rovers
www.cyberrovers.com

Software Project Plan

I. Table of Contents

1.1 Goals and Objectives

The main purpose of WMITS is to help automate the entire process that the Department of Environmental Quality (DEQ) Waste Management Division (WMD) staff members perform throughout an inspection. The goals of WMITS are:

- To minimize the time span of any inspection
- To minimize the amount of paper work required
- To provide a searchable database of all past inspections
- To provide an automated channel for the public to request information (under Freedom of Information Act)

1.2 System Statement of Scope

1.2.1 General Requirements

The following general requirements were laid out for our project named WMITS:

- A way in which DEQ could add new facilities to the database.
- A way in which DEQ could generate electronic checklists.
- A search on all electronic checklists.
- A way in which they could generate letters to be sent out to facilities based on inspection results.
- A way in which all letters and checklists could be stored electronically.
- A way to search for existing facilities.
- A way to print blank checklists and staff reports.
- A way in which they could view data which was entered into the database prior to our software.
- DEQ wanted a product that would allow them to easily add new checklists and letters or change existing checklists and letters.

Critique: Bounding is a critical element of the project scope and the project plan. It would be a good idea to try to "bound" all basic requirements (e.g., size of the database; estimated number of letters to be generated; types/number of staff reports)

- **Interface Enhancements**
  Staff members of WMD have requested a lot of interface enhancements that will increase the usability of the product for the staff.

- **Database Administrative Interface**

There is currently no documented interface for WMD staff members to maintain the checklist and letter templates. Should no such interface existed, Cyber Rovers will have to implement one from scratch.

- **Online Help**
    To develop an extensive help menu for the users and also to setup the online help for the need of the help in the future.

- **Training**
    The staff members have also requested throughout training for the entire staff for use with the software.

Critique: Use of word such as "lot of" and "extensive" are open to broad interpretation and therefore misunderstanding.

We will also implement a web-based help desk for WMD staff members to report bugs and request support. The helpdesk will be located at *http://www.cyberrovers.com.*

1.2.2 Extended Enhancement

Palm Pilot Integration

Out of the two extended enhancement requests (palm pilot integration & online record access), the team and client both agree on doing the palm pilot integration. From the design point of view, online record access has a major security risk, which the team has little or no experience on it. Palm pilot integration on other hand, needs only long programming, which can be (and will be) achieved by the team. We also suggest to the DEQ that they can make the online record request to be the next semester's project.
Before we decide on what kind of Palm Pilot we use, the team and the client have explore several options.

Comment: An implied risk is noted in blue above. This should be further addressed in the risk management section of the plan.

Database Restructuring

The current database structure is not optimized at all. We will try to improve it as we go along.

1.3 System Context

Eventually, multiple users will be using the product simultaneously. Therefore, concurrent connection will be an issue for implementation. In addition, this is a pilot product that hopefully, if successful, can be used in other locations as well. This leads to issues about future support for a larger user base.

1.4 Major Constraints

Time

We only have about two months to finish all documentation, software creation and enhancements. We have a lot of ideas but cannot implement them due to time constraint. One of the major ones is to move the application to be completely browser-based.

Funding

To develop and implement the Palm Pilot integration, we will need funding to buy at least one Palm Pilot. We will request the funding from University of Michigan – Dearborn should we decided to pursue this function.

2.0 Project Estimates

This portion of the document provides cost, effort and time estimates for the project using various estimation techniques, which will be elaborated in the appropriate section.

2.1 Historical Data Used for Estimates

Although this project is to enhance the existing software, we were unable to obtain cost information from the previous project team. However, average labor rate information is available.

We obtained the following data according to the InformationWeek Research's ongoing national IT salary survey:

Sector	: Government	Organizations
Job Function	: Application Development	
Experience	: Four years or less	
Company Size	: Below 100 Employees	

Low	US$62,000
Median	US$62,000
High	US$62,000

Low is the salary paid at the 25th percentile of all respondents in this data set; Median is the 50th percentile; high is the 75th percentile.

Data: InformationWeek Research 1999 National IT Salary Survey of 22,000 IT Professionals, survey period from February 1999 to November 1999.

Definition: Applications Development
Designs, specifies, codes, tests, debugs, maintains, and documents computer programs. May work with and modify packaged applications. Works with users in the support of business applications. May help users identify problems and design integrated solutions.

Since two of the members of this project are not experienced in Visual Basic, we will apply a -20% correction to the median salary. For this project, the estimated labor rate per person-month according to historical data will be:

$$\$((62{,}000 / 12)(0.80 + 0.80 + 1)) / 3 \approx \$4{,}500$$

Note: Members roles will be discussed in Section 5.0 Project Team Organization.

Comment: The above approach is completely acceptable in this context, although the team did not add in burdening (overhead), which would likely double the labor rate. For industry projects, local cost data MUST be used.

2.2 Estimation Techniques Applied and Results

Two estimation techniques have been used to generate two independent results for higher accuracy.

- Process-based
- Lines of Code (LOC) → COCOMO Model

2.2.1 Process-Based Estimation

For process-based estimation, the process is decomposed into a relatively small set of activities or tasks. Then, the effort required to accomplish each task is estimated. Based on the project scope, the following software functions are defined:

- User Interface Re-engineering   UIR
- Database Re-engineering   DR
- Database Administrative Interface   DAI
- Existing Bug Fixing   EBF
- PalmPilot Integration   PI

Activity →	Cust. Comm.	Plan-ning	Risk Analysis	Engineering		Construction Release		Cust. Eval.	Totals
Task →				Analysis	Design	Code	Test		
Function									
UIR	0.40	0.02	0.02	0.02	0.50	0.30	1.00	0.05	2.31
DR	-	0.01	0.10	0.10	0.30	0.10	0.10	-	0.71
DAI	0.20	0.01	0.05	0.05	0.40	0.20	0.06	0.05	1.02
EBF	0.20	0.01	0.02	0.01	0.02	0.50	0.08	0.05	0.89
PI	0.25	0.02	0.04	0.20	0.50	0.30	1.00	0.06	2.37
Total	1.05	0.07	0.23	0.38	1.72	1.40	2.24	0.21	7.30

% effort	14.38	0.96	3.15	5.21	23.56	19.18	30.68	2.88	100

Table 1 – Process-based Estimation Table

Based on the historical data we obtained, the estimated effort is approximately **7.3 person-months** and the estimated project cost is $4500 x 7.3 ≈ **$33,000**.

Comment: Note that the team allocated 14% of project eddort to initial customer communication. Commendable!

2.2.2 LOC-Based Estimation

The following estimates are based on "best-effort" estimation from previous programming experiences and existing software size.

Functions		Estimated LOC
User Interface Re-engineering	UIR	2,300
Database Re-engineering	DR	200
Database Administrative Interface	DAI	1,000
Existing Bug Fixing	EBF	800
PalmPilot Integration	PI	1,000
Total Estimated Lines of Codes		5,300

The estimates for LOC are plugged into the COCOMO formula for effort and duration estimation. The basic COCOMO model is used, for which

- Effort $E = a \, KLOC^{b}$
- Duration $D = c \, E^{d}$

The project is classified as an organic project, using default values a = 2.4, b = 1.05, c = 2.5 and d = 0.38.

$$E \quad = 2.4(KLOC)^{1.05}$$
$$= 2.4(5.3)^{1.05}$$
$$\approx 14 \text{ person-months}$$

$$D \quad = 2.5E^{0.38}$$

$$= 2.5(14)^{0.38}$$
$$\approx 6.8 \text{ months}$$

N $= E/D$
   $= 14/6.8$
   ≈ 2

Above results indicate that for two staff members, it will take 6.8 months to finish the project. Since we have three team members, the project duration should be shorter. Our best-effort estimation for the project duration because of the additional team member is **3 months**. Based on that calculation, the estimated project cost will be $4500 x 3 x 3 ≈ **$40,000**.

Comment: The team used the COCOMO model. Acceptable, but the new COCOMO II model would be a better choice today. Also, the results of the two estimation approaches are unusually close to one another. Don't expect this for every project.

2.4 Project Resources

2.4.1 People

This project will require three programmers in order to be finished in time. Each of the members will have to have specific skills (either obtained previously or on the fly). Team members will have to work in an interrelated network environment (ego-less) where everyone needs to communicate with everyone else on the regular basis.

Critique: Obtaining specific skills "on the fly" is a way of life in software projects and often can't be avoided. However, it does introduce risk. In an ideal setting, the team should budget time for training and/or experimentation required to obtain necessary skills.

2.4.2 Minimal Hardware Requirements

Development
Three IBM PC or compatibles with the following configurations
- Intel Pentium II 333MHz processor
- 64MB SDRAM
- 500MB Hard disk space
- Internet Connection

User Client-side
IBM PC or compatibles with the following configurations

- Intel Pentium 166MHz processor
- 32MB SDRAM
- 20MB Hard disk space

User Server-side

IBM PC or compatibles with the following configurations
- Intel Pentium II 333MHz processor
- 64MB SDRAM
- 500MB Hard disk space

2.4.3 Minimal Software Requirements

Development

- Windows 98
- Windows NT Workstation
- Windows NT Server
- Visual Basic 6.0 (three user licenses)
- Microsoft Access 97 and 2000
- Microsoft Word 97 and 2000

User Client-side

- Windows 98/NT Workstation
- Microsoft Access 97/2000 (optional)
- Microsoft Word 97/2000 (optional)

User Server-side

- Windows NT Server
- Microsoft Access 97/2000 (optional)
- Microsoft Word 97/2000 (optional)

3.0 Risk Management

3.1 Scope and intent of RMMM activities

We want the software to be free of any defects or errors, but it is hard or at times almost impossible to develop a system that is free of any defects. To be safe we would like to have a risk management plan to counter any difficulties that may impact the development or the creation of the software. Our goal is to assist the project team in developing a strategy to deal with any risk. For this we will take a look at the possible risks, how to monitor them and how to manage the risk.

For software development to avoid any risk both the developer and client have to work together. Client has to spend time with the developer in the beginning phase of the software development. If client decides to change the software, meaning if client wants to add some more functions into the software or to change the requirement, this will have major effect on the development of the software.

Critique: Avoiding risk is a noble goal, but risk is inevitable. The real focus of risk management is to understand the risks that are likely to occur and develop plans for dealing with them.

3.2 Risk management organizational role.

Every one associated with the software has responsibility of managing the risk. That is if everyone participated and paid close attention to all the details during the early phase of the software development many risk can be avoided.

- Software development can avoid having risk by double-checking their schedule, product size, estimates regarding costs of the development etc.
- Customer can help avoid risk by providing all necessary software information during the early phase of the development.
- Software development team can avoid risk by getting all the details of the equipment that are provided or are accessible to them.
- Client can avoid risk by making all necessary business changes before initiating request for the software.

Cyber Rovers
www.cyberrovers.com

3.3 Risk Description

This section describes the risks that are likely to be encountered during this project.

3.3.1 Description of Risks

Business Impact Risk:
This is the risk where concern is that of the not being able to come up or produce the product that has impact on the client's business. If the software produced does not achieve its goals or if it fails to help the business of clients improve in special ways, the software development basically fails.

Customer Risks:
This is the risk where concern is client's motivation or willingness in helping the software development team. If the client fails to attend meeting regularly and fails to describe the real need of the business the produces software will not be one that helps the business.

Development Risks:
If client fails to provide all the necessary equipment for the development and execution of the software this will cause the software to become a failure. So in other words customer has to be able to provide time and resources for the software development team. If all the requested resources are not provided to the software development team odds for the software development to fail rises greatly.

Employee Risk:
This risk is totally dependent on the ability, experience and willingness of the software development team members to create the working product. If the team members are not experience enough to use the application necessary to develop the software it will keep pushing the development dates until it's too late to save the project. If one or more members of the software development team are not putting in all the effort required to finish the project it will cause the project to fail. Employee risk is one of the major risk to consider while designing the software.

Process Risks:
Process risk involves risks regarding product quality. If the product developed does not meet the standards set by the customer or the development team it is a failure. This can happen because of the customer's failure to describe the true business need or the failure of the software development team to understand the project and than to use proper equipment and employees to finish the project.

Product Size:

This risk involves misjudgment on behalf of the customer and also the software development team. If the customer fails to provide the proper size of the product that is to be developed it will cause major problems for the completion of the project. If software development team misjudges the size and scope of the project, team may be too small or large for the project thus spending too much money on project or not finishing project at all because of shortage of finances.

Technology Risk:
Technology risk involves using technology that already is or is soon to be obsolete in development of the software. Such software will only be functional for short period of time thus taking away resources from the customer. Since the technology changes rapidly these days it is important to pay importance to this risk. If customer request use of software that soon to be obsolete software development team must argue the call and have to pursue customer to keep-up with current technology.

3.4 Risk Table

The following table describes the risks associated with the project. The appropriate categories of the risks are also given, as well as probability of each risk and its impact on the development process.

Probability and Impact for Risk m

The following is the sorted version of the above table by probability and impact:

Category	Risks	Probability	Impact
Employee Risks	Lack of training and experience	40%	1
Process Risk	Low product quality	35%	1
Product Size	Where size estimates could be wrong	30%	2
Development Risks	Insufficient resources	30%	2
Customer Risk	Customer may fail to participate	20%	3
Technology Risk	Obsolete technology	10%	2
Business Impact	Product may harm the business	10%	3

Table 1 - Risks Table (sorted)

Impact Values	Description
1	Catastrophic
2	Critical
3	Marginal
4	Negligible

Above is the table that categorizes the risks involved in software development. It gives brief description of the risk in Risks column and also provides the probability of risk occurring in percentages in Probability column and also the impact of the risk in the Impact column.

The impacts values assigned to the each risk are described in the section below the risk table. It is very convenient way to look at the risk and derive the information of the risk.

Critique: It is important to provide a reference to the RMMM document (which does exist elsewhere in the project docs.). Even if the RMMM Plan has not as yet been developed, the Project plan should provide an indication that it will be developed and a provisional pointer to it.

4.0 Project Schedule

Following is the master schedule and deliverables planned for each stage of the project development lifecycle, and their respective planned completion dates.

4.1 Deliverables and Milestones

Stage Of Development	Stage Completion Date	Deliverable	Deliverable Completion Date
Planning	01/21/00	Quality Assurance Plan Project Plan Milestone	01/15/00 01/20/00 01/21/00
Requirements Definition	02/25/00	Draft Requirements Specification Draft Design Specification Project Test Plan Requirements Specification (final) Milestone	02/09/00 02/09/00 02/15/00 02/22/00 02/22/00
Design (Functional & System)	03/01/99	Draft Training Plan Program and Database Specifications Design Specification (final) Milestone	02/23/00 02/26/00 02/29/00 03/01/00
Programming	04/02/00	Software (frontend and backend) System Test Plan User's Guide Operating Documentation Milestone	03/31/00 03/10/00 03/20/00 03/28/00 04/02/00
Integration & Testing	04/15/00	Test Reports Training Plan (final) Acceptance Checklist User's Guide (final) Milestone	04/03/00 04/10/00 04/14/00 04/12/00 04/15/00
Installation & Acceptance	04/20/00	Maintenance Plan Acceptance Test Report Milestone	04/16/00 04/20/00 04/20/00

4.2 Work Breakdown Structure

Please refer to the above table for the work breakdown structure (deliverables are set according to logical work breakdown).

Cyber Rovers
www.cyberrovers.com

Comment: In most cases, it would be a good idea to complement this section with a more detailed tool-based project schedule description using, say, MS Project.

5.0 Project Team Organization

The structure of the team and the roles of team members are defined in this section.

5.1 Team Structure

Due to the small size of the project team, the team will be organized in an egoless structure, where the entire group will make most of the decisions together.

Conceptual and Advanced Interface Development

- Overall process specification
- Database re-engineering
- Advanced Interface Development
- Internal Modules Development
- Draft Documentation

User Interface Design and Development / Trainer

- Intermediate User Interface Development
- Training Sessions
- Operational Manual
- Draft Documentation

Editor / Master Tester / Maintenance

- Proof Reading
- Overall Testing and Reports
- All Final Documentation
- User Guide

Note: The above responsibility descriptions are only used as a guideline only. During the course of development, a team member will most likely required to help in another area because of the small team size.

5.2 - Additional Member Responsibilities

The following are additional notes on team members' responsibilities:

- Each person will work on multiple tasks throughout the development process in addition to their main role.
- Due to the small size of the team, project design and specification will be discussed with the whole team.

- All development personnel are required to prepare draft documentation of their work, as well as individual testing on their part. Tester will do the overall final testing at the end of the testing stage.
- One programmer will take on the role as coordinator to ensure the project is on schedule, as well as scheduling In-Stage Assessments.

6.0 Tracking and Control Mechanisms

6.1 Quality Assurance Mechanisms

- Tight Change Management
- Extensive before implementation Design using Rapid Prototyping
- Close Contact with Clients, meeting every two weeks and regular email contacts
- Plenty of Research on PalmPilot platform before development

For complete information please refer to the document titled *Software Quality Assurance*.

Critique: The team does not mention the use of formal technical reviews as a QA mechanism. Because FTRs are one of the more important QA techniques, they should be noted here as well as in the SQA Plan.

6.2 Change Management and Control

- For changes affect the user experiences we will have to notify all clients
- For changes that do not affect the user experiences we will notify a client representative
- Due the size of the team, internal control panel will be used. One member of the team suggests a change, it will need to be approved by the other two members
- Formal version numbering will be used. All version changes must be documented in a common document accessible to all team members before a new version number can be released. Version number will be structured as follows:

```
<Major Release>.<Minor Release><Bug fix>
```

Version changes will be reviewed during ISA's. Not only the previous version, but also all older versions of any document or codes will be preserved. This will ensure if the need to revert back to more than one version arises, the necessary version is available.

For complete information please refer to the document titled *Software Configuration Management Plan*.