SOFTWARE PROJECT MANAGEMENT PLAN

Project Enlightenment

Computer Training For Visually Impaired Automation Tool (CTVIAT)

Prepared By: Project Enlightenment

Date: October 21, 2010

Version No: v2.1

Document Change Control

The following is the document control for revisions to this document.

Version Number	Date of Issue	Author(s)	Brief Description of Change
1.0	21/10/10	Project Enlightenment	Phase 1 - Preliminary draft for Phase 1

Definition

The following are definitions of terms, abbreviations and acronyms used in this document.

Term	Definition
CTVI	Computer Training for Visually Impaired
RAD	Requirements Analysis Document
CTVIAT	Computer Training for Visually Impaired Automation Tool
JFW	JAWS for Windows
SDD	Software Design Document
ODD	Object Design Document
CPU	Central Processing Unit in a Computer

Table of Contents

1. OVERVIEW

This plan is for development of the Enable India product by a small group, whose name is Enlightenment, of senior standing computer engineering students consisting of five individuals: Emmar, Eray, Ferhat, Osman, and Özge.

The Enable India product is to develop a software product to support the Enable India (EI) that helps visually impaired people. First of all, this project is a term project of software engineering course. Secondly, this project is a registered SCORE project and Kapil Wasmani is the advisor from Microsoft Research.

1.1 Project Summary

1.1.1 Purpose of the System

This paper describes the details of Computer Training for Visually Impaired (CTVI) Automation Tool. The CTVI enables visually impaired students to take some teaching courses followed some training exercises on the computers to test their knowledge at Enable India. Since process of correcting the exercises is currently done manually by trainers, it is an exhaustive and time consuming task. The Automation Tool we present here automates this process. Its main purpose is to make the training program more productive, efficient and accurate.

By automating the training process, the tool also aims to make evaluation process easier, faster and more secure.

From an admin or a trainer's perspective, the tool purposes to reduce the burden of controlling, correcting and pursuing the testing process. Also, the tool allows creating tests in different modules of instructions(such as editing, word, excel) and in different types(e.g., objective, descriptive or action-based), adding questions to question bank, designing specific question papers to perform examination by both giving question specific assistance to help finish their tests and correcting the exercises when the student fails to handle a question.

1.1.2 Scope of the System

The CTVI Automation Tool is implemented for two kinds of users: vision-impaired students and CTVI trainers (or admins). Those students who are capable of using a computer with the help of Braille-based tools are the primary users of Automation tool. These users require no other additional computer experience.

The CTVI administrators are the main beneficiaries of the Automation Tool. The tool performs all the tasks the trainers perform manually today. What admins are required to do is just login to system and use the tool and benefit from the facilities of the tool.

The tool will be designed such that all the user related properties will be accessible by any user and she will require minimum knowledge when using the tool. Through this, users can receive more personalized services and obtain relevant information with relatively less effort and time.

1.1.3 Objective and Success Criteria

Security (or a secure platform) is one of the most important success criteria. Since users log in to system with their own accounts, secure maintenance of user-specific data is required.

System reliability which means system services will function correctly is also an important success criterion. If a system is unreliable, it is difficult to ensure system safety or security also. Another important success criterion is scalability which means system should cope with new demands by adding new resources. This can be handled through careful design and implementation of Automation Tool.

1.1.4 Assumptions and Constraints

Constraints include the following:

- The deadline must be met.
- The budget constraint must be met.
- The product must be reliable.
- The architecture must be open so that additional functionality may be added later.
- The product must be user-friendly.

1.1.5 Project Deliverables

Last version of the project is supposed to be an automation tool which provides computer training for visually impaired people. The complete product, including user manual, will be delivered until the deadline of the project submission.

1.1.6 Schedule and Budget Summary

The duration, personnel requirements, and budget of each workflow are as follows:

- Requirements workflow(4 weeks, five team members, 2400 TL)
- Analysis workflow(25 days , five team members, 7680 TL)
- Design workflow(8 days, five team members , 3840 TL)
- Implementation workflow(39 days, five team members, 14866TL)
- Testing workflow(25 days, five team members, 4800 TL)

The total development time is 65 days, and analysis, design, implementation and testing workflows will go on together because of chosen software development life cycle and preselected staffing. The total internal cost is absolutely 29376 TL. The details are in open project file.

1.2 Evolution of the Project Management Plan

All changes to the project management plan must be agreed to by Kapil and course instructor before they are implemented. All changes should be documented in order to keep the project management plan correct and up to date.

2. Reference Materials

All artifacts will conform to the company's and course's programming, documentation and testing standards.

Extra artifacts are Project Enlightment.pod which is produced by open project for detailed schedule and cost estimation, tasks.pdf gives detailed information for tasks such as well defined name, resources, estimated completion time with start and finish date, predecessors and reference id, whoiswho.pdf gives mapping between team members and tasks and answers who does what question.

3. Definitions and Acronyms

EI Enable India, our client.

SCORE a competition in software engineering.

Roll back is to choose a reference point to return back and to return back this stable point when trainee gave a false answer to give hint and one more try.

Hint is to give guidance trainee to solve problems easily.

GUI is short version of Graphical User Interface.

Objective, Descriptive and Action-based are the types of questions in automation tool.

DAO means Database Access Object.

IDE is integrated development environment.

4. Project Organization

4.1 External Interfaces

All the work on this project will be performed by the members of the group Enlightenment. All will meet weekly with Kapil and course instructor to report progress and discuss possible changes and modifications.

4.2 Internal Structure

The development team consists of Emmar, Eray, Ferhat, Osman and Özge.

4.3 Roles and Responsibilities

All group members will participate in all phases of development cycle but each group member has special features so each member will do much of work in her specialized task. Therefore, Özge firstly will be responsible for design, so testing is Emmar's and implementation; control and documentation are Eray, Ferhat and Osman's.

Details are in whoiswho.pdf.

5. Managerial Process Plans

5.1 Start-up Plan

5.1.1 Estimation Plan

As previously stated, the total development time is estimated to be 65 days and the total internal cost to be 29376 TL. These figures were obtained by software management tool, namely open project. Details are Project Enlightment.pod file.

5.1.2 Staffing Plan

All team members are needed for the entire 65 days because everybody will participate and we tried to utilize from resources perfectly.

Details are in references. However, below table gives medium details of development.

	Documentation	Design	Analysis	Implementation	Testing
Eray	20	10	18	39	18
Emmar	30	8	15	19	25
Ferhat	21	8	18	42	18
Osman	20	8	18	40	18
Özge	30	8	15	19	25

5.1.3 Resource Acquisition Plan

All necessary hardware is already available except Braille based keyboard which will be used to test system. All inputs from Braille keyboard is just like normal keyboard so tests will be run on normal keyboards. However, if project can be done before deadline, artifacts will be delivered to Enable India for them to test in real environment. Indeed, the only other required hardware is a computer which runs Windows 7. Each group member has qualified computer. Microsoft tools are chosen for development tools which can easily downloaded from DreamSpark. Chosen tools are Visual Studio as IDE (only fully featured development environment for C#), Microsoft SQL Server as database (perfect integration with Visual Studio) and Microsoft Office for design, presentation and documentation (Word, Excel, PowerPoint and Visio). Moreover, a repository is needed which will be supplied by university. Therefore, a SVN client is needed and TortoiseSVN client is chosen because it perfectly integrates with windows explorer and has simple usage. Project also requires Word and Excel APIs in which turns .NET. Some useful CASE tools areNCover and NCoverCop for line by line code coverage and to notice where is untested, NUnit and MbUnit for unit testing which are successful ports of JUnit, and DotNetMock for again unit testing which is a framework and library to facilitate the use of Mock Objects and has perfect integration with NUnit and MbUnit, Systin for System testing, which is a port of Systir from Ruby to C# and allows users to specify plain English text software requirements that can then become executable tests, log4net for logging which is perfectly logging tool from Apache and is familiar to developers. Final remark is that http://www.google.com/microsoft.html is also will be very useful because organization and presentation of Microsoft's materials are always timely and handy. All necessary hardware, software, and CASE tools for the project are already available. The product will be delivered to Enable

India as a desktop application and also some scripts will be supplied to configure a database server at local network.

5.1.4 Project Staff Training Plan

Firstly, software development life cycle methodologies are studied in websites suggested by instructor.

Then, software configuration management is a new concept to team members to work on so http://code.google.com/intl/tr-TR/edu/tools101/scm.html is very useful.

Database is the very important part of the project so http://code.google.com/intl/tr-TR/edu/tools101/mysql.html may b e useful also.

For GUI, PRO WPF in C# 2010 Matthew MacDonald can be useful.

Some references and books can be needed for C#.

Training of Visio (documentation), open project (management plan) and NUnit, etc (testing and code coverage) is also required.

5.2 Work Plan

5.2.1 Work Activities and Schedule Allocation

2 days	Project planning
8 days	Design phase
11 days	Initial implementation(DAO, system requirements)
10 days	Most important function implementation and testing
15 days	Control and report
21 days	Additional(extendable) functions (hint and rollback)and testing
5 days	GUI and user manual

Some part will be performed parallel; details are in Project Enlightment.pod file.

5.2.2 Resource Allocation

Firstly, software development life cycle is determined. Extreme programming is selected as methodology; general principles are explained below table and causes as the following:

CTVIAT is not hardware originated project and even purely software originated except Braille tools. This eliminates waterfall methodologies. CTVIAT is not large and expensive project, so not spiral life cycle. Cowboy coding is not formal enough for this software competition. Agile and XP are very similar indeed. However, XP is chosen because 1) XP is more relax, 2)Group has two developers who can easily program in pair, 3)Group has one designer and one reviewer(documenter) who can review progress, 4)Group has one ambitious tester who can do unit testing of all code, 5) Group has one responsive advisor whom group can easily communicate for progress, code, design and changing requirements,

6) Program domain is not familiar to staff(Braille for developers), 7)We will code for Word automation first and delay Windows Explorer, Excel, etc. In the beginning, we will put empty functions for these functionalities, after Word automation is completely functional, we will go on them.

Waterfall	Spiral	Agile	XP	Cowboy
-Big Design Up Front -Sequential manner -Used for hardware originated and stable problems -Changes costly -Bug fix in requirements -Easily markable milestones	-Relaxed waterfall -Initial phase BDUF -Second phase prototype, review and improve on incremental(design-code in small motivations) -Usually used in large, expensive and complicated projects -Chosen in game development -Agile is chosen in smaller projects	-Relaxed spiral -No long-term planning and minimal planning in short -Lack of planning, requires small increments -Each increment involves full sdlc -Changes are welcome -Customer satisfaction by rapid development -Continuous delivery(Live code) -Live code is measure of progress -Coop business and dev -Face-to-face communication -Self organizing team	-Some is good, extreme is better(best) -Introduce new principles on top of agile -Programming in pairs -Extensive code review -Unit testing of all code -Implement when needed -Simple and clear code -Expect changes in requirements(via customer and developer) -Frenzy communication	-No external management -Lack of formal methodologies -Decrease burden of bureaucracy -Free working -Student level(inexperienced developers and experimental projects) -Chosen as hobby by talented developers -Quick and dirty, code and fix implementation (unreadable source and conflicts in semantics)

The two team members will work separately on their assigned artifacts and three developers will work together. Osman's assigned role will be to monitor the daily progress of the group, oversee implementation, be responsible for overall quality, interact with the client and help implementation. Team members will meet after three days and discuss problems and progress. Formal meetings with client will be held at the end of each week to report progress and determine if any changes need to be made and meetings with instructor will be held at scheduled timeline by the syllabus. Osman will ensure that schedule and budget requirements are met. Risk management will also be Osman's responsibility. Osman also has overall responsibility for all documentation and has to ensure that it is up to date. Actually, Osman is the manager. Design is Özge's responsibility and testing is Emmar's and

implementation is controlled by Eray and Ferhat. Maximizing user-friendliness and functionalities is Özge's and minimizing faults Emmar's top priorities.

5.2.3 Budget Allocation

The budget for each workflow is as follows:

Requirements workflow 2400 TL

Analysis workflow 7680 TL

Design workflow 3840 TL

Implementation workflow 14866 TL

Testing workflow 4800 TL

Total 29376 TL

5.3 Control Plan

Any major changes that affect the milestones or the budget have to be approved by Kapil and course instructor and documented.

No outside quality assurance personnel are involved. The benefits of having someone other than the individual who carried out the development do the testing will be accomplished by each person testing another person's work products.

Osman will be responsible for ensuring that the project is completed on time and within budget. This will be accomplished through three to meetings with team members. At each meeting, Eray and Ferhat will present period's progress and problems, Özge will present design of next increment and Emmar will present test result of old increments. Osman will determine whether they are progressing as expected and whether they are following the specification document and the project management plan. Any major problems faced by the team members will immediately be reported to Osman.

5.4 Risk Management Plan

The possible risk factors that can be faced during project developments are as follows:

The customer, visually impaired, although is familiar with using computers during training; she was getting help from a trainee to complete the testing process. When using our product the customer will be alone and use the computer by herself. Therefore our product should have a user-friendly interface.

The product that we are developing is software rather than hardware. During developing the product we will has no chance to test the product with proper hardware (just use our own computers). However, visually impaired customer will use the product with a Braille-based tool and there is a risk of failure driven by hardware. The feedbacks taken from Mr. Kapil enable us to decrease the risk of such hardware failures.

Our product will need storage of some information (such as questions, answers, etc.) and the user should have reached necessary info as fast as possible during training. These will be handled through careful design and implementation of the product.

During the development of the project every team members will be assigned a different task of the project and work on that part. During the testing, developers should compile their code together with other team member's code. There is a risk of incompatible code parts that leads to failure in testing.

During weekly meetings these issues will be covered and the optimal solutions will be argued to decrease the probability of the risk.

5.5 Project Close-out Plan

No planning because project cannot be withdrawn must be implemented in every possibility.

6. Technical Process Plans

6.1 Process Model

The Extreme Programming will be used.

6.2 Methods, Tools and Techniques

The workflows will be performed in junction with the Extreme Programming. The product will be implemented in C#.

6.3 Infrastructure Plan

The product will be developed using .NET 4.0 running on Windows 7 on a personal computer.

6.4 Product Acceptance Plan

Acceptance of the product by our client will be achieved by an automation tool that satisfies perfectly all requirements on time.

7. Supporting Process Plans

7.1 Configuration Management Plan

New versions of software systems are created as they change: For different machines/OS; Offering different functionality; Tailored for particular user requirements. Configuration Management aims to control the costs and effort involved in making changes to a system. In CTVIAT project configurations may be about the addition of the new types of questions all products of the software process namely specifications, designs, programs, test data and user manual have to be managed with an international standard such as IEEE standard for CM. However, in case of a database arrangement all products of the software process do not have to be managed.

7.2 Testing Plan

The testing workflow of the Extreme Programming will be performed. All code is tested by unit tests.

7.3 Documentation Plan

Documentation will be produced as specified in the Extreme Programming, SCORE and course requirements. Moreover, task assignment is balanced so is implementation phase because each member will participate in implementation and document what she implemented.

7.4 Quality Assurance Plan

Quality Control aims to identify whether a product meets its specifications or not and to ensure that the results generated by the testing are correct. For CTVIAT, since customers' quality requirements are efficiency, reliability, security, etc and project developers' quality requirements are maintainability, reusability, completing the project on time within the budget, quality control will be made through the weekly meeting of the team members. During these meetings project manager control whether the project development progress continue and project management plan is followed as expected or not. If a problem is identified, such as latency, insufficient budget, etc, the project manager will be responsible to solve this issue with team friends. The quality assurance will be concerned with whether the right test is carried out and right results are delivered to right person at the right time or not. Mr. Kapil will be informed at each step of project development and the team will develop the product according to reviews with him and feedbacks taken back such that project meets the customers' needs. Also quality documentation will be a record of the progress and support continuity of development if the members of team changes.

7.5 Reviews and Audits Plan

Every member will write some functionality and document her function specification. Then members will exchange their functions and test these functions according to written spec. Moreover, major test such as integration and system tests will be carried out by Emmar and Özge.

7.6 Problem Resolution Plan

Any major problems faced by team members will immediately reviewed by Mr. Kapil and instructor. Moreover, if any problem is faced in implementation, design can be relaxed and changed.

7.7 Independent Verification and Validation

Two principle objectives of V&V process are to discover of defects in the system and the assessment of whether or not the system is useful and useable in an operational situation. Therefore, to correctly produce the product with minimum number of defects we may need a V&V process at each stage of the software process. In the producing process static verification will be useful with the help of a code analyzer such as FxCop, StyleCop and Gendarme for C#. At the milestones and at the end of the production dynamic verification will be useful in which the system is executed with test data and its operational behavior is observed. Development process and V&V process are dependent with each other because the way followed in development process sometime will be determined by V&V process if there is a defect or a wrong operation during the execution.

7.8 Subcontractor Management Plan

7.9 Process Improvement Plan

8. Additional Plans

Additional components:

Milestones:

CTVIAT will be a teacher which asks questions that are broadly classified into 3 types

- Objective
- Descriptive
- Action-based

Objective type questions are multiple choice questions where the student is given 4-5 answer options from which student can choose the right answer.

Descriptive type questions have descriptive answer which cannot be compared by program for its accuracy.

Action – based questions require the student to perform a specified task on his computer.

In first milestone, we will provide a live code that satisfy general requirements,

- Exercise topics can be separated into modules (editing, word, excel, etc.)
- Support for mp3 files as instruction and answer.
- Create specific question types(Objective, Descriptive, Action-based)
- The time taken is calculated
- Load old exam
- Repeat old exam

And also accessibility requirements those are keys to answer questions. Moreover, Unicode format and individual login are supported. To supply, general requirements we will provide question preparation template for descriptive and objective type questions.

In the second milestone, we will start to support template for action—based questions related to editing, dialog, windows explorer, jaws help, word, excel, and other requested applications. Test control function will be completed. With this function, admin can generate individual reports based on each individual user's performance.

In final version, CTVIAT will give hint students and also if necessary or requested by student like another try in a game which probably will entertain students, rollback with some minus points in action-based questions. Very user-friendly GUI also will be working.

CTVIAT will make lives of teachers and students much easier.

Security:

A password can be needed to use the product.

Training:

No training is planned because client is very far from developers but very detailed user manuals will supplied to client. Moreover, if client requests so, some guidance can be done by tele-conference on Skype.

Maintenance:

Corrective maintenance or enhancement can be performed voluntarily because this project has social responsibility but this has not been planned or requested yet.