ICE @ Georgia Tech: AP Practice Exam

Project Plan

Version 3.0 Final
26 July 2005

CS3911 Team II

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 01	Lisa Jordan Daniel Osiecki	Preliminary information entered in all sections	7 June 2005
Draft 02	Daniel Osiecki	Update after environment choice.	20 June 2005
Final 03	Daniel Osiecki	Update after inital grading and feedback.	26 July 2005

1 Introduction 1

1.1 Overview 1
1.2 Deliverables 1
1.3 Assumptions and Constraints 1
1.4 Reference Materials 1
1.5 Definitions and Acronyms 1

2 Management Structure 2

2.1 Project Lifecycle 2
2.2 Project Organization 2
2.2.1 External Interfaces 2
2.2.2 Internal Structure 2
2.2.2.1 Roles and Responsibilities 2
2.2.2.2 Staffing 3
2.3 Communication 3
2.4 Risk and Asset Management 3

3 Planning and Control 5

3.1 Estimate 5
3.2 Resource Identification 5
3.2.1 Staff 5
3.2.2 Time 5
3.2.3 Cost 5
3.2.4 Materials 5
3.3 Resource Allocation 6
3.3.1 Milestones 6
3.3.2 Work Breakdown Structure 6
3.3.3 Schedule 7
3.4 Tracking and Control 7

4 Technical Process 8

4.1 Engineering Environment, Methods, Tools, and Techniques 8
4.2 Development Environment, Methods, Tools, and Techniques 8

5 Supporting Plans 9

5.1 Configuration Management 9
5.2 Quality Assurance/Testing 9

1Introduction

1.1Overview

The Ice @ Georgia Tech AP Practice Exam is an existing online system that aits students and teachers in studying by providing a practice environment for the AP Computer Science (A/AB) Exam. The purpose and objective of this project is to extend upon the existing AP Practice Exam system, adding new functionality to help students and teachers receive greater benefits from using the system and improve the system as a whole by addressing performance and stability concerns. New functionality will include features such as the ability to log in, save exam results, and track student statistics.

1.2Deliverables

The system will be comprised of the following elements to be delivered:

- Full system source code
- Complete database structure and content template as plaintext SQL
- Webserver configuration files to support provided source code
- Revised Software Design documentation including Database Design

Delivery will take the form of CD-ROM media and an operative site install, if server access can be provided for the latter.

1.3Assumptions and Constraints

This project is an extension of previous work, and has established expectations. These expectations must be preserved during redevelopment and enhancement of the system. This includes existing program functionality and appearance.

The software system will be based upon the programs and data already in place, though both are subject to change during the development provided functionality and appearance are preserved.

This development will specify requirements and work that are out of the scope of the 11-week timeframe of development. As a result, not all features suggested will be implemented. Requirement documentation will specify which features are to be included in the delivered system and which will not appear. As many features as possible will be implemented, as prioritized by the sponsor, within the constraints of the prescribed 15 to 18 work-hours per week per developer.

1.4Reference Materials

All reference materials provided by the sponsor are located at the sponsor's web site at the time of project commencement:

http://coweb.cc.gatech.edu/ice-dev/47

1.5Definitions and Acronyms

AP - advanced placement
CVS - concurrent versioning system
ICE - Institute for Computing Education
SDS - software design specification
SRS - software requirements specification
Swiki - a collaborative online website

2Management Structure

2.1Project Lifecycle

This project will follow a Staged Delivery model, as this best supports the constraints on the developers. The project milestones listed herein will be delivered one at a time, and the content of each milestone will not be subject to change unless approved by the sponsor and all team members.

After initial requirements gathering and project planning are complete, the development will focus on both a robust design and a subsequent implementation. A subset of all requirements gathered will be designated for design and implementation. A large product of this project is a well-documented, formal design for use in demonstrating design concepts. A final set of extended functionality will be specified but not implemented.

The project will be considered complete when the required software system has been delivered. No provisions for system maintenance are expressed or implied.

2.2Project Organization

The project will be staffed by a four-person student team, responsible to a faculty sponsor.

Project Sponsor: Barb Ericson
Group Leader: Daniel Osiecki
Testing Lead: Chase Peeler
Design Lead: Lisa Jordan
Development (Implementation) Lead: Chad Hansen

2.2.1External Interfaces

Major contact with the sponsor will take place during scheduled weekly meetings. These meetings will be scheduled at the preceding meeting or via electronic mail and may or may not require the attendance of all team members. All team members required to attend will be given no less than 24 hours of notice about the meeting, and no required meetings will be scheduled during a student's registered class hours.

Other communication with the sponsor will be carried out via electronic mail as described in section 2.3.

2.2.2Internal Structure

2.2.2.1 Roles and Responsibilities

The following roles are a general guideline for the responsibilities of each group member. It is expected that all team members may take on responsibilities beyond those listed here, respecting their primary goal of completing all project deliverables.

Role	Responsibility
Project Manager	Communicates with sponsor, finalizes formal documents, communicates with project administrators (i.e. professor and TAs).
Requirements Lead	Communicates with sponsor to elicit requirements, organizes final set of requirements into SRS
Design Lead	Organizes and distributes design work, finalizes and checks all design documents
Development/Implementation Lead	Organizes and distributes development work, compiles and deploys overall software system
Development Engineers	Implements software application and database

Role	Responsibility
Test Lead	Organizes, assigns, and supervises testing tasks

2.2.2.2 Staffing

Role	Staff Member	Start Date	End Date
Project Manager	Daniel Osiecki	3 June	28 July
Design Lead	Lisa Jordan	3 June	23 June
Requirements Lead	Daniel Osiecki	3 June	23 June
Testing Lead	Chase Peeler	8 Jul	28 July
Implementation Lead	Chad Hansen	24 June	28 July
Development Engineer 1	Lisa Jordan	24June	28 July
Development Engineer 2	Daniel Osiecki	24 June	28 July
Development Engineer 3	Chase Peeler	24 June	28 July

2.3Communication

Contact with the customer will occur through weekly meetings, as described in section 2.2.1. The customer may also contact the project lead through email or the swiki. Group members will communicate through email and meetings scheduled as necessary.

2.4Risk and Asset Management

The following risks have been identified from the outset of the project, and mitigation strategies are provided herein:

2.4.1 Legacy Code

The existing code at the outset of the project does not meet the standards set for performance and organization. This is in part due to choice of language, and must be addressed as a top priority. The preferred solution is to reimplement existing functionality in duplicate using a superior language for the circumstances. If this is not possible, the following risk (2.4.2) will take priority. The alternative solution is to improve the organization and coding practices of the existing implementation as much as possible, at a sacrifice of features.

2.4.2 Unstable, Inefficient Environment

If the legacy environment must be maintained (see above), the primary risk will be the unreliability and poor performance of the programming environment. Jakarta/Tomcat and JSP are not considered suitable for real-time programming, and as a result, high-load systems suffer from poor performance. This platform is also known to contain critical bugs that are poorly documented. Finally, native Database support in this environment does not implement reliability features. As database communication is important to the software, this is a high risk.

Mitigation strategies for this situation will involve the performance of more strict testing procedures, and possible sacrifice of best coding practices for the sake of performance.

2.4.3 Unfamiliar Environment

If the legacy environment must be maintained (see above), a secondary risk will be the obscurity of the programming environment to the team. JSP and Servlet configuration are not straightforward and require more complex configurations than alterna-

tives. If this environment must be used, it is recommended that one team member become responsible for configuration management, to mitigate this risk.

2.4.4 Evolving Requirements

Because this development is not time-critical and does not have formal requirements set forth by the sponsor, motivation towards a successful implementation may be at risk. Further, requirements may evolve or "creep" because of this. The mitigation strategy has been to attempt to formalize requirements as much as possible from what the sponsor has initially recommended.

2.4.5 Imprecise Time Estimation

Although all team members are confidently capable of achieving project goals, lack of experience may hinder ability to properly estimate time required to do so. As a result, setting time-based goals and milestones is difficult. This risk has been mitigated by prioritizing requirements and familiarizing the sponsor with the situation to an understanding that progress will be monitored for forward accomplishment on all tasks, but that the achievement of all tasks is at some risk.

2.4.6 Inability of Team Members to Meet Time Requirements

Because team members are all university students, time allocation is expected to be unpredictable and frequently below expectations. Time will be tracked precisely and publicly viewable along with frequent correlation with the project plan to identify slipping goals as soon as possible, in hopes of motivating effort.

3Planning and Control

3.1Estimate

Time estimation for the development of this system is being performed by dividing the design and development into tasks that may be completed somewhat atomically, dependent on one another in a linear fashion. This will allow assessment of project progress despite the relative difficulty of estimating the time required to complete individual requirements due to complex dependencies between requirements.

Initial estimates will be made based solely upon the experience of the developers. Time spent on each set of features and requirements will be documented precisely to gage the accuracy of the first estimates and revise later-phase estimates if acceptable and required.

3.2Resource Identification

The resources available to the team will be constant over the project lifecycle. These resources include:

- Project Background - the initial requirements for the project, along with the legacy implementation of the system and system data.
- Sponsor Feedback - a very active cycle of feedback and planning is expected for the duration of the project.
- Team Contributions - a consistent amount of time is expected to be available from all team members for the duration of the project.
- Facilities - computing facilities will take the form of personal computers managed by each team member as well as class-provided facilities. The latter will likely not be utilized to any great extent. The target hardware system is expected to be made available to developers for the minimum purpose of delivering the software system and testing on the target platform.
- Communication Tools - the sponsor has provided a Swiki for inter-team and team-sponsor communication. Team members have access to advanced electronic mail distribution capabilities which have already been put into use.

3.2.1Staff

No outside staff is expected to work on this project.

3.2.2Time

The timeframe for this project is exactly 12 weeks, inflexible, due to class constraints. An expectation of 15-18 hours/week/person has been required by class administration.

3.2.3Cost

There is no financial budget for this project. The only commodity to be expended here is team member time. Administrative restrictions have dictated 15-18 hours of work/week/person. This constitutes over 600 hours of time to be devoted to all tasks, including design. This cost is flexible, however, as up to 120 extra hours total may be requested from team members, for the sake of meeting goals. Team members may, at their own discretion, exceed these time estimates.

3.2.4Materials

No specialized materials or equipment will be required for this project. All expressed software platform requirements are freely available and free of charge.

3.3Resource Allocation

3.3.1Milestones

Milestone	*Date*	*Description*
M1	20 May	Team Formation Complete
M2	27 May	Project Assigned, Roles Assigned
M3	27 May	Introduction to Sponsor
M4	8 June	SRS, Project Plan Due
M5	14 June	Environment Choice Argument Cutoff
M6	8 July	SDS
M7	26 July	Functionality
M8	27 July	Revised Documentation
M9	4 August	Delivery

3.3.2Work Breakdown Structure

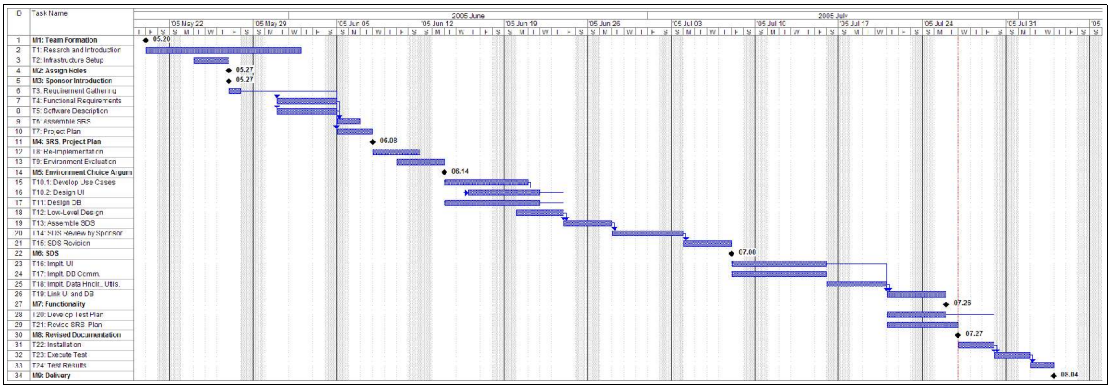
Task	*Prerequi-sites*	*Team Member*	*Time Estimate (hours)*	*Description*
T1		Daniel	20	Research project, communicate with sponsor to set up meetings.
T2		All	5x4	Set up team communications, email, swiki.
T3		All	2x4	Gather requirements from sponsor.
T4	T3	Daniel	15	Develop functional requirements.
T5	T3	Chad, Chase	5x2	Software Description for SRS
T6	T4, T5	Lisa	10	Assemble and complete SRS
T7		Daniel	15	Develop Project Plan, set up hour tracking
T8.1		Chase, Chad	15x2	Redevelop legacy functionality
T8.2	T8.1	Daniel	20	Perform, document comparative benchmark of legacy, redeveloped systems
T9	T8	All	2x4	Develop arguments for environment choice
T10.1		Lisa, Chase, Chad	10x3	Develop use cases
T10.2	T9, T10.1	Chase, Chad	15x2	Design User Interface
T11	T9	Daniel	20	Design Database and Data Structures
T12	T9	Daniel	20	Design low-level system components

Task	*Prerequi-sites*	*Team Member*	*Time Estimate (hours)*	*Description*
T13	T10, T11, T12	Lisa	10	Compile, complete SDS
T14	T13	All	10x4	Review design with sponsor
T15	T14	Daniel	20x1	Revise SDS around sponsor recommendations
T16	T15	Lisa	40	Implement Graphical User Interface
T17	T15	Chad, Chase	40x2	Implement exam data structures and DB commnication interface.
T18	T16, T17	All	10x4	Implement page data hander system.
T19	T16, T17, T18	All	10x4	Link UI and DB backend together.
T20		Chase	15	Develop Test Plan
T21		Daniel, Lisa	30x2	Revise all documentation for regrade
T22	T19	Daniel, Chase, Chad	10x3	Installation of system on sponsor hardware
T23	T22	All	15x4	Execute Test Plan
T24	T23	Daniel, Lisa	15x2	Compile Test Results

An additional 5 hours/person/week will be allocated for team meetings, sponsor meetings, and administrivia.

3.3.3Schedule

The following schedule outlines the approximate timeframe for the milestones and tasks listed above.



3.4Tracking and Control

All work changes will be reported to the team leader. Hours will be recorded weekly on the team swiki. Changes to the project schedule will be given by the team leader via email.

4Technical Process

4.1Engineering Environment, Methods, Tools, and Techniques

The following tools will be used for ERM and UML diagrams during this project:

- Microsoft Visio
- Poseidon

The Following tools will be used for document preparation:

- Microsoft Word
- Open Office
- Microsoft Project
- Microsoft Visio

The following tools will be used for communication between group members:

- email (Dedicated CS3911 mailing list)
- phone
- AIM

4.2Development Environment, Methods, Tools, and Techniques

All development will take place on personal computers, the specifications of which are irrelevant to the scope of this project.

The following tools will be used for coding:

- Textpad
- Vim

Unit testing will be performed with PHPUnit.

Code documentation will be created with PHPDocumentor.

5Supporting Plans

5.1Configuration Management

This project will be utilizing a code repository similar to CVS to ensure organization and change control. The developmental manager will be responsible for setting up this repository. The details of repository location and format will be determined once a design document is in place.

5.2Quality Assurance/Testing

As milestones in this project are reached, they will be demonstrated to the sponsor, who will have the option of approving or requesting revisions. In this way we will ensure the quality of the features and the satisfaction of the customer. Unit testing will be carried out incrementally, as milestones are reached. As the format of the tests will depend largely on the design of the project, for more information please reference the Test Plan, to be completed at a later date, once a design document is in place. This plan will include a final test of operational functionality and performance.