

ACSS

Version 1.1
Feb. 10th, 2006.

Team #2

2COMMUNICATE

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Jeffrey Qua	Initial Draft	Feb. 6 th , 2006
1.1	Jeffrey Qua	Final Draft	Feb. 10 th , 2006

Contents

1 INTRODUCTION.....	3
1.1 OVERVIEW.....	3
1.2 DELIVERABLES.....	3
1.3 ASSUMPTIONS AND CONSTRAINTS.....	3
1.4 REFERENCE MATERIALS.....	3
1.5 DEFINITIONS AND ACRONYMS.....	4
2 MANAGEMENT STRUCTURE.....	5
2.1 PROJECT LIFECYCLE.....	5
2.2 PROJECT ORGANIZATION.....	5
2.2.1 External Interfaces.....	5
2.2.2 Internal Structure.....	5
2.2.2.1 Roles and Responsibilities.....	5
2.2.2.2 Phase Responsibilities.....	6
2.3 COMMUNICATION.....	6
2.4 RISK AND ASSET MANAGEMENT.....	6
3 PLANNING AND CONTROL.....	8
3.1 RESOURCE IDENTIFICATION.....	8
3.1.1 Staff.....	8
3.1.2 Time.....	8
3.1.3 Materials.....	8
3.2 RESOURCE ALLOCATION.....	9
3.2.1 Milestones.....	9
3.2.2 Work Breakdown Structure.....	9
3.2.3 Schedule.....	9
3.3 TRACKING AND CONTROL.....	10
4 SUPPORTING PLANS.....	11
4.1 CONFIGURATION MANAGEMENT.....	11
4.2 QUALITY ASSURANCE.....	11

1 Introduction

This introduction section will provide the context for this project.

1.1 Overview

ACSS is a platform-independent educational tool that delivers a learning experience through a game format. The purpose of this software is targeted towards high-functioning autistic children who are able to communicate. While they are able to converse, they have trouble understanding proper times to interrupt conversations between other people. Children can play and improve their skills in understanding proper times to interrupt conversations. The game-based approach will encourage users to learn in an engaging and interactive experience. This software is being developed for an untested field of possible solutions for autistic children with normal social skills but do not comprehend the nuances of conversation leading to a breakdown in communication.

The game will be made to be a simple mental and motor exercise where the child is able to acknowledge when they think it is an appropriate time to speak. This focuses the scope of the product towards addressing issues with conversation skills only.

1.2 Deliverables

Several documents will be delivered: the Specific Requirements Specification document, the Systems Design Document, and the User Manual.

The software delivery component will consist of source code, the content delivery module, game system (includes the tutorial and demonstration features), as well as files that comprise the game(s) such as video files, timing files, etc.

The software should be able to be installed, or copied over from CD to hard drive.

1.3 Assumptions and Constraints

The project schedule will be 11 weeks or less, starting from January 24, 2006 and ending on March 27th, 2006 by the latest. There are six members in the team. Each team member will be responsible for approximately 12 hours of work each week. There will be approximately 1000 man-hours allocated to this project at the maximum. The software must be able to deliver video and audio content according to the game specifications (i.e. inside the game interface).

1.4 Reference Materials

(To be updated in next revision)

1.5 Definitions and Acronyms

ACSS - Autistic Conversational Skills Software

User – any person who uses the program

Administrator – a person who has administrative privilege/access to the system

GUI – graphical user interface

Main Menu Page – the initial menu to allow users to register/login

Game Menu Page – the initial menu for users who have logged in

Registration Page – the page that allows the user to create new accounts

Login Page – the page that allows the user to login if they have a valid login

UML – Unified Modeling Language. Refer to <http://www.uml.org/>

OS – operating system

SRS – Software Requirements Specification

SDS – Software Design Specification

RC – Release Candidate

CVS – Concurrent Versioning System

2 Management Structure

This section covers all of the aspects of managing this project that are not related to estimating, planning, and controlling of the work.

2.1 Project Lifecycle

We shall use a single iteration waterfall model as the lifecycle model for this project. We will be performing the phases of the project in the following order:

Phase	Expected Completion Date
requirements analysis	Feb. 10 th , 2006
system design	Feb. 24 th , 2006
program design	Mar. 7 th , 2006
coding	Mar. 7 th , 2006
unit and integration testing	Mar. 21 st , 2006
system testing	Mar. 24 th , 2006
acceptance testing	Mar. 27 th , 2006

We may be unable to accomplish acceptance testing as we do not have immediate access to our targeted user domain.

2.2 Project Organization

The project team currently consists of six members. The team will consist of one Project Manager who will oversee the progress of the project. However, each member will also take on a different sub-management role, as well as acting phase leaders for the different stages of the project lifecycle.

2.2.1 External Interfaces

Our software development team has three external interfaces. Our Communication & Contact Manager, Wei Lin, will be the primary conduit of communication to these resources, unless alternate contact is appropriate. The three external interfaces are: the client, Dr. Pat Mirenda; the project supervisor / course professor, Kurt Eiselt; and the teaching assistant assigned to our group, Izzet Safer.

2.2.2 Internal Structure

2.2.2.1 Roles and Responsibilities

Management Role	Responsibilities	Manager
Project Manager	Project Overseer and Planner	Jeffrey Qua
Minutes Manager	Records minutes of meetings	Dan Liu
Communication & Contact Manager	Primary communications representative	Wei Lin

Configuration Manager	Ensures completion in software development phases and that requirements are satisfied	Dan Liu
Progress Manager	Reports on progress of project	Michael Tien
Research & Training Manager	Researches and informs team of any required technical or domain-based knowledge	Wei Lin
Risks Manager	Assesses risks and any potential problems	George Firican
Software Version Control Manager	Organizes any issues with Version Control Management	Ian Cook
Webmaster	Maintains the team website	Jeffrey Qua

2.2.2.2 Phase Responsibilities

Phase	Phase Description	Phase Leader
Requirements Elicitation	Develop System Requirements Specification document	Wei Lin
System Design	Develop System Design Document	Jeffrey Qua
Program Design	Develops the program	Ian Cook
Coding	Implement the software	George Firican
Unit & Integration Testing	Test the software and integrate different units	Michael Tien
System Testing	Test the software system as a whole	Michael Tien
Acceptance Testing	Perform testing with user domain	Dan Liu

2.3 Communication

Internal communication within the group is primarily through e-mail, supported by updates through the website. Team meetings are scheduled for every Tuesday between 2pm and 3pm in X330 in X-Wing of the ICICS building.

2.4 Risk and Asset Management

- 1) Java vs Other programming languages
 - We have planned our approach to the design problem by implementing in Java
 - Video-playing issues
 - Cross-platform compatibility (Platform independence)
 - Java is well documented
- 2) Time
 - Every member has heavy loads of work; time will become an issue
 - Hard deadlines are required
 - Have to develop software as well as media content.
 - Recording video and audio.
 - Writing scripts.
 - compile timing settings for each video.
- 3) Learning
 - Need research / tutorial sessions regarding video libraries in Java, timing issues, accounts, account recording

4) Team Members

- Workload is distributed evenly among members and will shift depending upon each person's ability to handle a certain task to prevent anyone from being overburdened
- The Project Manager will deal with conflicts amongst members
- The risk of members dropping is unpreventable/unpredictable and any responsibilities of members who leave will be distributed among the rest of the members

3 Planning and Control

3.1 Resource Identification

3.1.1 Staff

For each project phase, each member will be assigned specific responsibilities for the current tasks. Phase leaders will be in charge of distributing the task assignments for their respective phases. The Progress Manager will inform the team on the development progress throughout the software life cycle. The Project Manager will have the final call in determining any decisions requiring executive judgment.

3.1.2 Time

The project timeline will start as early as possible in January and will continue until the end of the allotted time period on March 27th, 2006.

3.1.3 Materials

Computer System Requirements

- PC and/or Mac based OS

Media Creation Tools

- Video Camera
- Adobe Photoshop
- Avid / Final Cut Pro
- Audio Software

IDE (Java)

- Eclipse

System Design Tools

- Poseidon for UML
- Microsoft Visio
- Microsoft Project

3.2 Resource Allocation

3.2.1 Milestones

Milestone #	Description	Due Date
1	Management roles & phase leaders assigned	Tue Jan 17 / 06
2	Website functional & running	Thu Jan 19 / 06
3	SRS Rough Draft	Tue Jan 31 / 06
4	Team name established	Tue Jan 31 / 06
5	Project / Product Title established	Tue Jan 31 / 06
6	SRS First Pass	Mon Feb 06 / 06
7	Project Plan Draft	Mon Feb 06 / 06
8	SRS Final Draft	Fri Feb 10 / 06
9	SDD High-Level Design Completed	Wed Feb 15 / 06
10	SDD First Draft Completed	Mon Feb 21 / 06

3.2.2 Work Breakdown Structure

Phase leaders are responsible for assigning and distributing work load among different members. Specific task assignments may be shared internally between team members when necessary. Subgroups may be created in specific phases to address complex tasks. These subgroups will be determined when deemed appropriate.

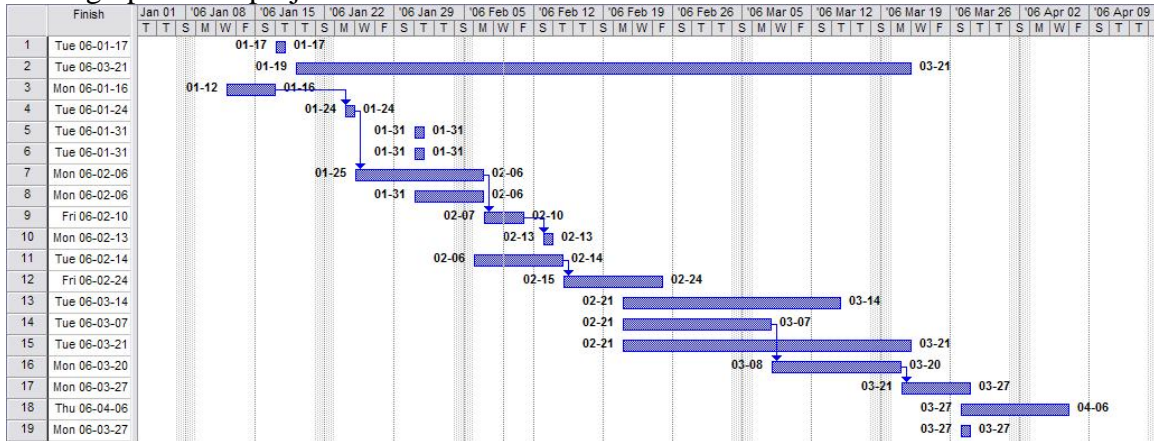
3.2.3 Schedule

Primary Task Leads are in **bold**

	Description	Start	End	Assigned to
1	Management roles & phase leaders assigned	Tue Jan 17/06	Tue Jan 17/06	-
2	Website designed & functional	Thu Jan 19/06	-	Jeffrey
3	Establish client meeting date	Thu Jan 12/06	Mon Jan 16/06	Wei
4	Client meeting	Tue Jan 24/06	Tue Jan 24/06	-
5	Team name established	Tue Jan 31/06	Tue Jan 31/06	-
6	Project / Product Title established	Tue Jan 31/06	Tue Jan 31/06	-
7	SRS First Draft	Tue Jan 24/06	Mon Feb 06/06	All, Wei
8	Project Plan First Draft	Tue Jan 31/06	Mon Feb 06/06	Jeffrey, Dan
9	SRS Final Draft	Mon Feb 06/06	Fri Feb 10/06	All, Wei
10	SRS signed off by client	Fri Feb 10/06	-	Wei
11	SDD First Draft	Mon Feb 6/06	Tue Feb 14/06	Jeffrey, Ian
12	SDD Final Draft	Tue Feb 14/06	Fri Feb 24/06	Jeffrey, Ian
13	Implementation / Coding	Tue Feb 21/06	Tue Mar 14/06	All, George
14	Test Plan First Draft	Tue Feb 21/06	Tue Mar 07/06	All, Michael

15	Testing Phase	Tue Feb 21/06	Tue Mar 21/06	All, Michael
16	Test Plan Final Draft	Tue Mar 07/06	Mon Mar 20/06	All, Michael
17	User Acceptance Testing	Tue Mar 21/06	Mon Mar 27/06	Dan
18	Delivery	Mon Mar 27/06	Fri Apr 07/06	All
19	Presentations	Mon Mar 27/06	Fri Apr 07/06	All

Gantt graph of the project schedule:



3.3 Tracking and Control

The overall project schedule and control will be closely observed by the Project Manager and will step in if any major delays or problems arise. Otherwise, phase leaders will have responsibility during phase periods for keeping the team on track and on schedule.

Implementation control will be handled by the Software Version Control Manager to organize and direct with CVS. Any problems in implementation will be reviewed and managed by the Coding Phase Leader or any sub-manager who was assigned to that coding task.

Individual accomplishments and personal work hours will be reported to and reviewed by the Progress Manager who will then forward the information to the Webmaster for work record updates on the website.

Any major decisions during meetings will be recorded by the Minutes Manager who will then forward the data to the Webmaster for updating decision documentation on the team website.

If any executive or design conflicts evolve between team members, the Project Manager has veto authority for judgment calls.

4 Supporting Plans

4.1 Configuration Management

The SRS document, SDS document, and software code are the artifacts for this project. CVS will be used for the change control of all implementation artifacts, and any minor changes will result in version updates on the CVS. All team members can make changes to any code, except for the front-end user interface code. The SDS document will define the user interface, and will not be modified unless all team members approve of the change. If any team member wants to change the user interface, that member must bring the topic to discussion at a team meeting and explain why the change is necessary.

There will be daily builds updated on the CVS until all features have been implemented and when there are no major bugs found. A release candidate (RC) build will then be created and be used for system and acceptance testing. The software will be released after the system passes all the testing phases (approved by all team members), and the documentation will be released upon the phase leader's approval.

4.2 Quality Assurance

The schedule in this Project Plan document, section 3.2.3, will be followed to ensure that work is done at a secure rate. The phase leader and the progress manager will contact all teammates regularly to ensure that everyone is on schedule.

If there is any bug found during the testing process, a “bug track” must be issued with the following parameters: issue number, name of the tester, date, title, a brief description of the bug, test environment (i.e. what kind of OS and version), and status (active, pending, and closed). All the bug tracks shall be archived on the CVS server that will be available to all the team members. If a bug has been fixed and tested, the bug track must be updated with the tested date, name of the tester, and the status of it shall be changed to “closed”.

Daily builds will be available to ensure the quality of the system. Each daily build will be tested to ensure the functionality of the system. If any problem with the daily build arises, the phase leader for the unit/integration/system testing will inform all team members.

A test plan outline has been created with the approximate time for each testing phase. It is outlined as follow:

Unit test (1 – 1.5 weeks)

This shall be done with JUnit Test Suite. Each function in the code shall be tested with an expected output (each team member will be assigned to test certain areas). A JUnit test function would return true if the actually output matches the expected output; otherwise, it would return false.

Integration test (1 – 1.5 weeks)

The “**Bottom-Up**” testing model is likely to be used to achieve the integration test; however, this might be changed during the testing process.

System test (1 – 1.5 weeks)

The tests shall be done on both Windows and Mac. Since there is only very limited amount of time, tests might only be run on Windows 98 (or 2000) and Mac 10.2 (the oldest edition of OS requested by the client). The system test phase is broken down into the follow components.

- Test cases creating

Test cases shall be created with simple description and test procedures. This can be done with an excel sheet and added to the CVS archive so that each team member will be able to mark the test result directly in the file.

- Functional testing (1-2 days)

Each team member will be assigned to test certain operations. According to our SRS document, we have the following operations:

- Register
- Log-in
- Demo
- Tutorial
- Practice
- Recorded score review
- Settings category
- Video category and difficulty
- Volume and keyboard settings
- Video learning session
- Bonus learning session
- Reward
- Score output per current session
- Log-out
- Exit

- Dry run (2-3 days)

This is the first run of the test. Every test case shall be run, and the bugs found shall be reported and fixed.

- Second run (2-3 days)

This is the second run of the test. Every test case shall be run, and the bugs found shall be reported and fixed.

- Regression test (1-2 days)

This is the final run of the test. Only the test cases which were failed during the second run shall be run, the bugs found shall be reported and fixed. If there are any bugs that cannot be fixed after the regression test, they shall be reported in the user document.

Performance testing (1-2 days)

This shall be done in order to check if the performance of the system fulfills the performance requirements specified in the SRS document.

Installation testing (1 day)

This shall be done in order to determine if the software system can be successfully installed and run on various platforms specified by the client.

The software code will be reviewed once the implementation is completed. The review will consist of evaluation of subsystems and functions code to ensure correct implementation. The code will also be subjected to a requirements check to ensure all requirements are met.