Jarvis Emulator

Project Management Plan

COP 4331, Fall 2015

Modification History

Version	Date	Who	Comment
v0.0	9/13/2015	Robin Schiro	Created document
V1.0	9/15/2015	Jimmy Lam	Added Project Team Organization
V2.0	9/15/2015	Jimmy Lam	Added Technical progress metrics and plan for tracking, control and reporting of progress
V2.1	9/17/2015	Manuel Gonzalez	Added Quality Assurance, Table of Work Packages and PERT Chart

Team Members:

- Jimmy Lam
- Julian Rojas
- Manuel Gonzalez
- Robin Schiro

1) Project Overview

a) This application uses facial detection and recognition to respond to users as they enter or exit a room. A webcam is set up to point toward the room's entrance. When a user enters, the program can respond with an action that has been configured by that user. The GUI of the application will allow users to set up profiles that specify how the program should behave. Based on this configuration, the application will communicate back and forth with the user to perform desired actions and provide useful information. If more than one person is in the room at once, the application will perform actions based on who is talking at any given time.

2) Reference Documents

a) Concept of Operations

3) Applicable Standards

a) **Coding Standard**

i) Generally, we should follow the standard defined by Microsoft for .NET programming here. However, there are a few exceptions and additions:

(1) Do not use the 'var' variable type. It is always best to be as unambiguous as possible when declaring variables. Using 'var' only increases the chance of potential confusion.

b) **Document Standard**

i) Normal Font: Calibri Body, 11

ii) Headings:

(1) Document Title: Calibri Body, 14, Centered at top of document

(2) Major Section: Calibri Body, 14, Bold

(3) Subsection: Calibri Body, 11, Bold

iii) Spacing: Place a line break between every major section of the document

iv) Table of Contents: Include a Table of Contents if a document is longer than 10 pages.

(1) Format: The heading of each major section should be left aligned and the corresponding page of each major section should be right aligned. There should be a line a periods separating each heading from its corresponding page number.

v) Modification history: A table with columns labeled Version, Date, Who, and Comment will be placed at the top of every document to display the document's modification history. Because the table has a 'Who' column, we do not need to maintain a separate list of authors' names.

c) **Artifact Size Metric Standard**

i) Our team will use Size Points to determine the difficulty/amount of work involved in accomplishing a task. A Size Point (SP) corresponds to a realistic amount of time that any one of us might spend on the project in one day: 2 hours. 1 Size Point can be equivalent to 2 hours of work by a person with our average amount of experience. For example, a class that would take the average person on our team 8 hours to write would be measured at 4 SP.

4) Project Team Organization

a) Our group members are Robin Schiro, Manuel Gonzalez, Julian Rojas, and Jimmy Lam. We will take part in all aspects of the project, however, we will each focus on a feature of the project. Robin will be our project manager.

b) **Responsibilities:**

i) Robin –input from camera and algorithm for facial recognition, user interface, documentation

ii) Manuel – Speech construction AI so that it talks to user, documentation

iii) Julian – Primary action module (APIs, weather related info, give info from relevant websites to user), submodule to detect person talking, documentation

iv) Jimmy – Speech recognition, other action modules (open/close applications, logging in/out), documentation

c) We will communicate in person as well as virtually through Google Hangouts if physical meetings aren't possible. We will meet at least once a week to discuss the responsibilities of each group member and to check on everyone's status and the status of the project.

5) Deliverables

Artifacts	Due Dates
Meeting Minutes	Emailed out by midnight on the day of the meeting
Individual Logs	Updated after each change to the code base
Group Project Management Reports	12/3/15
Concept of Operations	9/18/15
Project Management Plan	9/18/15
SRS	10/8/15
High-Level Design	10/29/15
Detailed Design	10/29/15
Test Plan	10/8/15
User's Manual	12/3/15
Final Test Results	12/3/15
Source, Executable, Build Instructions	12/3/15
Project Legacy	12/3/15

6) Software Life Cycle Process

a) Our group will adhere to a form of the Agile development process over the course of this project. This process divides development into a series of ‘sprints’ during which developers complete tasks that were assigned to them at the beginning of each sprint. Agile is very flexible due to the ability of customers and developers to report bugs and create user stories during each sprint. We would like to gain experience using this process because of its ubiquity in the industry of software engineering and its proven success.

b) We will have two-week sprints. At the beginning of each sprint, we will create several user stories and assign them to each team member.

c) Weekly Status meetings- Describe what you did during the week, problems you had, and what you plan to work on during the next week.

d) Diagram:

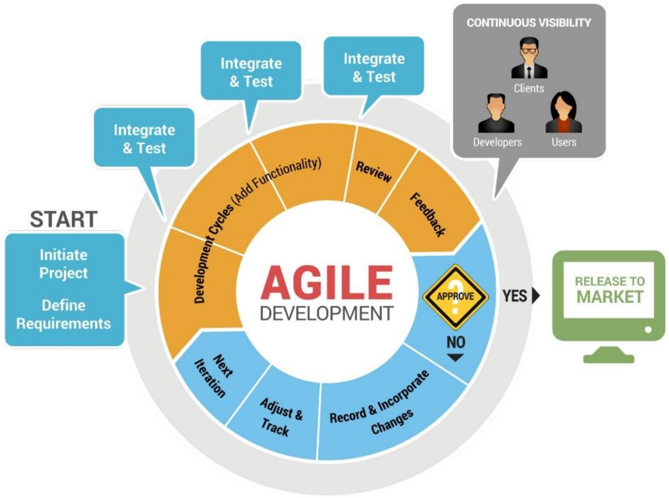


Figure 1: Representation of the Agile development cycle. Source: STAGroup

7) Tools and Computing Environment

a) The JARVIS Emulator will be programmed using Microsoft Visual Studio using the .NET framework. The language used to develop the program will be C# (an object oriented programming language). The Windows Presentation Foundation (WPF) will be used in the process of designing and implementing the GUI. This application will work on the Windows Operating System.

8) Configuration Management

a) **Version Control**

i) We will use Microsoft Visual Studio's Team Foundation Server as a repository for our code and documentation. Since we are building a .NET application, we are using Microsoft's Visual Studio as our IDE. Moreover, because Visual Studio is tightly integrated with the Microsoft's Team Foundation Server (TFS), we will be using TFS instead of Github as the host of our repository. Our repository will be used in conjunction with Git to manage checkouts and commits.

ii) Each member of the team will be responsible for minimizing conflict between pushes to the repository. Communication with the other members of the team is key in order to guarantee the repository's stability.

b) **Commit Process**

i) Before working on an issue, the developer must create a branch for that issue. This is done to prevent conflict when multiple developers are working on the project simultaneously.

ii) As the developer works on an issue, he should make several commits (depending on the size of the issue) to document the history of changes he makes to the relevant code files.

iii) Once the developer has resolved the issue, he should merge his local branch with the main branch. He must communicate with the rest of the team before performing a merge.

9) Quality Assurance

a) Several tests will be performed after each individual module meets the basic requirements for functioning. First starting with the common inputs that a module should expect on a normal use, and then using more extreme cases and multiple streams of data to test stress on each module.

b) After integrating the modules, tests should be performed on data transfer between the modules. Again, a big load of common data and some extreme cases will be used to guarantee the stability of the system.

c) Ultimately, general tests for overall performance will ensure user comfort and accessibility, as well as introducing multiple users to test accuracy when handling these types of situations.

d) Manuel will ensure to design the tests for each module and the whole system. He will perform the tests once each team member is ready, and any failed tests will be reported in Git's issue tracking.

10) Risk Management

a) Levels of expertise between team members will lead to redistribution of work. Balancing out workload will have to be a top priority in order to develop the program on time.

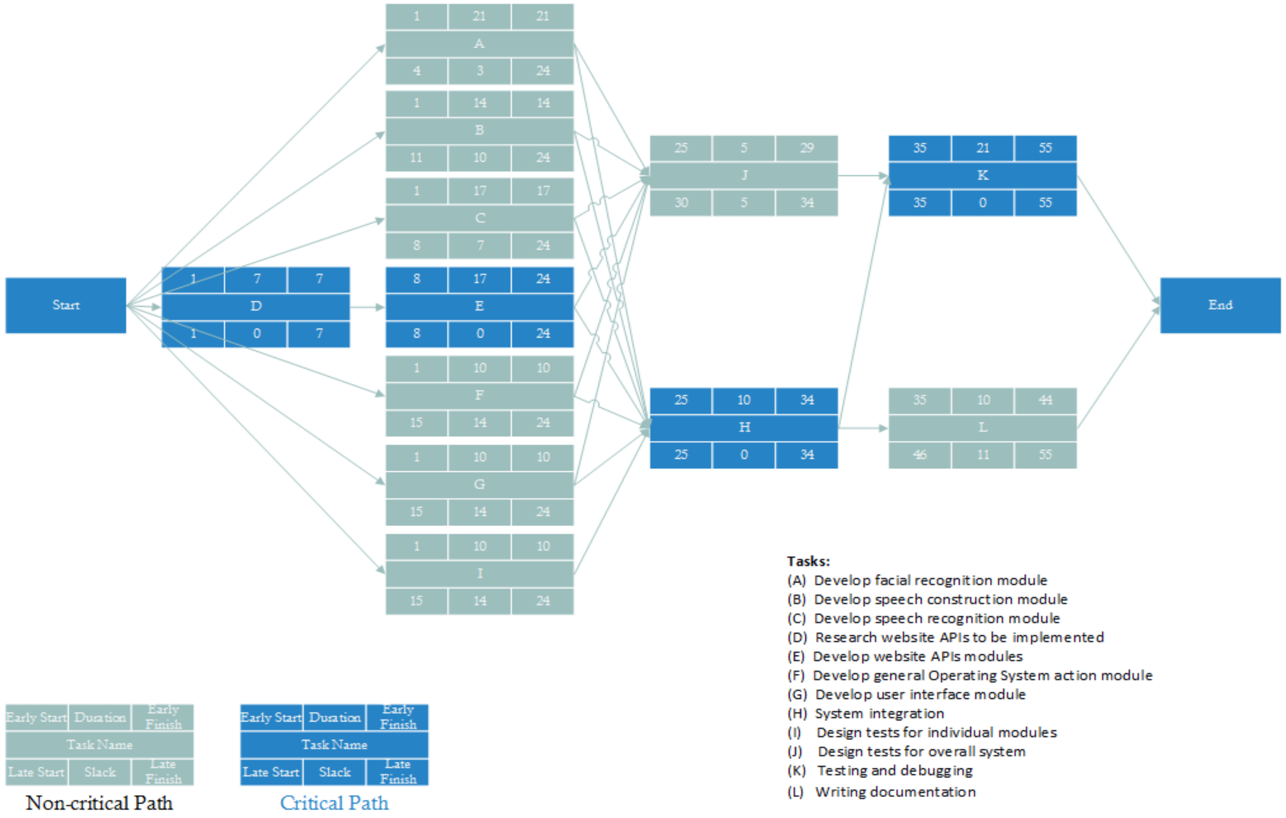
b) Application may be too intrusive for the user. We will need to find a way to deactivate some features that would bother some users (maybe remembering birthdays or calendar events) .

c) Security will be important, since certain information that the user may input on his or her profile might be of interest to phishing, targeted ads and redirected websites that are not the predetermined ones.

11) Table of Work Packages, Time Estimates, and Assignments

Work package	Time Estimate	Responsible
Develop facial recognition module	3 weeks	Robin Schiro
Develop speech construction module	2 weeks	Manuel Gonzalez
Develop speech recognition module	2 ½ weeks	Jimmy Lam
Research website APIs to be implemented	1 week	Julian Rojas
Develop website API modules	2 ½ weeks	Julian Rojas
Develop general Operating System action module	1 week	Jimmy Lam
Develop user interface module	1 week	Robin Schiro
System integration	1 week	Everyone
Design tests for individual modules and overall system	1 week	Manuel Gonzalez
Testing and debugging	3 weeks	Everyone
Writing documentation	3 weeks	Everyone

12) PERT Chart



Tasks:
(A) Develop facial recognition module
(B) Develop speech construction module
(C) Develop speech recognition module
(D) Research website APIs to be implemented
(E) Develop website APIs modules
(F) Develop general Operating System action module
(G) Develop user interface module
(H) System integration
(I) Design tests for individual modules
(J) Design tests for overall system
(K) Testing and debugging
(L) Writing documentation

13) Technical Progress Metrics

Metrics	**Measure**
Requirements	
Number of Requirements	6 requirements
Number of Requirement changes	
Specifications	
Number of Specifications	11 parts
Number of Specification Changes	
Analysis and Design Phase	
UML diagrams completed	
Implementation	
Number of Packages	
Number of Classes	
Number of Methods	
Number of Lines of Code	
Number of Test Cases	
Integration	
Number of Test Cases	
Number of Defects	

14) Plan for tracking, control, and reporting of progress

a) We will individually submit a weekly report to keep track of our own progress (number of hours working on the project, responsibility report), but we will also have weekly meetings to keep track of each other's progress and the status of the project overall. Our project manager will look at our progress reports, our technical progress metrics, reassess project risks, and our project plan to determine whether we are ahead, on time, or behind schedule. With this information, we will update our PERT chart, the technical progress metric, and update everyone with any changes to the project plan.

Jarvis Emulator
Concept of Operations
COP 4331, Fall 2015

Modification History

Version	Date	Who	Comment
v0.0	9/14/2015	Robin Schiro	Created document
v1.0	9/15/15	Robin Schiro	Updated details for 'Current System'
V2.0	9/15/2015	Jimmy Lam	Added 'Needs' and 'Impacts'
V3.0	9/16/2015	Jimmy Lam	Edited 'Needs' and 'Impacts'
V3.1	9/16/2016	Manuel Gonzalez	Added 'Users and Modes of Operation' and 'Analysis'

Team Members:

- Jimmy Lam
- Julian Rojas
- Manuel Gonzalez
- Robin Schiro

1) Current System

a) Currently, most PC users cannot identify themselves and interact with their computers using a system of facial recognition. In an environment in which one computer is shared between multiple people, a user might have to enter a password to log on. After that, he must focus on his monitor to open desired applications and visit websites that interest him. These activities require the user to physically input information into the computer.

b) Moreover, most desktop sessions do not allow multiple users to interact with the computer in the same setting. If the computer is "occupied" by someone, another person must wait his turn before taking advantage of the computer's resources.

2) The Proposed System: Needs

a) Firstly, our project needs the Windows operating system, as we will be coding a Windows desktop application. Our project requires a webcam for facial detection and recognition, as well as a microphone for users to perform voice commands. A language database is needed for 'Jarvis' to recognize the voice commands and construct sentences to respond to the user. We will need a front end GUI for the user to see when 'Jarvis' is speaking to improve the interaction level between the user and 'Jarvis'. The GUI will also be used to allow users to create a profile and configure settings with 'Jarvis' (for example, whether 'Jarvis' will log in when the user enters the room). Finally, we will need web APIs for 'Jarvis' to report information that is relevant to the user (based on user input in the GUI and based on which user 'Jarvis' detects).

3) The Proposed System: Users and Modes of Operation

The system will handle only one type of user, no specific roles will be present. Each user will be authenticated through facial recognition, this will allow the application to be customized depending on which user is interacting with it. The system should be able to handle multiple users at once. The system will perform a set of actions reacting to the user's behavior (leaving the room, speaking, not looking to the camera, moving, etc.), as well as executing commands detected through speech. Some actions may not be available for some users, depending on their profiles.

4) The Proposed System: Operational Scenarios

a) Normal Scenarios:

i) The user comes within the field of view of the camera; the application is able to detect a face. If the application has not previously been exposed to the users face, it will prompt them to complete a profile for it to be able to recognize them. Otherwise, it will greet them by their name.

ii) When a user is interacting with the JARVIS, it will be able to follow commands. For example, if the user starts a sentence with a keyword, it will take in a command that has already been established. The user says “JARVIS, open Microsoft Word” and the application will then answer “Opening Microsoft Word” as the new application loads.

iii) JARVIS will be able to detect multiple faces in the same frame, and it will be able to recognize them.

b) Atypical Scenarios

i) The most common one would be that the application would not be able to detect faces depending on the lighting or conditions of the room.

ii) Too many faces in the frame may cause the program to crash or confuse people’s profiles.

iii) Not understanding the commands given by the user or doing something completely different from what the user requested.

iv) A user that has previously interacted with the application comes into the field of view, and the application is not able to recognize him, therefore a second profile is created. Duplicate information leads to waste in computer’s memory.

5) The Proposed System: Operational Features

a) MUST HAVE:

i) The ability to recognize, distinguish and track people’s faces

ii) A GUI that allows the user to manually configure their own profiles, such as preferences, most used applications, most visited websites

iii) It must be able to perform speech recognition and construction

iv) The ability to report information from websites (such as weather)

v) The ability to open and close applications, and log in or out

b) WOULD LIKE TO HAVE:

i) The ability to analyze mouth movement from users to be able to determine whom it is interacting with

ii) An animated face to talk to the user

iii) Recognition of emotions and gestures

6) The Proposed System: Expected Impacts

a) **Enhanced Desktop Experience, Better Personalization, More Productivity**

When using the 'Jarvis Emulator,' users will have an enhanced desktop experience. Users will be able to open and close applications, and sign in and out of their computer simply through voice command and facial recognition. Not only that, 'Jarvis' will provide a personal experience with users as it will be able to detect who is talking and, if their faces are stored in 'Jarvis's' memory, address that person by name. If the face is not stored in memory, the new user can easily create an account with 'Jarvis', where it will change its behavior based on who is using it. 'Jarvis' will also help to improve user productivity; instead of users searching for weather or news, users can tell 'Jarvis' which websites they want information from, and 'Jarvis' will automatically update the users with all the information relevant to the users, saving them time to focus on their other daily needs.

7) The Proposed System: Analysis

Expected Improvements:

- The system should seamlessly integrate facial recognition, speech recognition and artificial intelligence.
- The system should feel like it "knows" the user.
- The user should be able to use our system without changing any of his/her usual routine.

Disadvantages:

- The user may feel uncomfortable of having a camera constantly watching what he/she is doing.

Limitations:

- It requires camera and a microphone to function.
- Only some online information will be available through the application's speech construction due to the difference in web API formats
- Due to the limited amount of information known about the user, the system may not be able to accurately predict some of the user's needs.

Risks:

- The current algorithm for facial recognition has some problems detecting specific users. We may not be able to fix such errors during our short development period.
- Many of the algorithms and techniques that we will be using are a bit out of the scope for undergraduate studies, and could be difficult to implement.

Alternatives and Tradeoffs:

- All algorithms that cannot be enhanced will be discussed for further research.
  Mainly, the Web APIs our system should support. Time constraints prevents us from having an extensive list of supported websites for the application to interpret and present to the user, so we will have to choose whether or not to implement specific websites. However, a system can be designed for add-ons and plug-ins that could increase the amount of websites supported.