

Team Automaton

# Project Plan

Fundamentals of System Engineering (EN.605.401.71.FA14)

Shambhavi Sankrit, Henohe Shiferaw, Ethan Wilansky  
9-23-2014

## Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) is broken down into a hierarchy similar to the Waterfall model. Each phase is further broken down into a set of deliverables. Smaller tasks serve as guidance to complete the larger tasks.

### 1. Clueless AI Project

#### 1.1 Planning

##### 1.1.1 Creating Team Charter

##### 1.1.2 Project Plan

###### 1.1.2.1 Create Work Breakdown Structure

###### 1.1.2.2 Create Vision for the project

###### 1.1.2.3 Create Project Schedule

###### 1.1.2.4 Create Risk Management plan

###### 1.1.2.5 Create Quality Plan

###### 1.1.2.6 Meet to finalize documentation

#### 1.2 Requirements Gathering

##### 1.2.1 Interface Specification

###### 1.2.1.1 Agreeing on interface Protocol

###### 1.2.1.2 Documentation showing all messages sent between subsystems

###### 1.2.1.3 Finalize documentation amongst group

##### 1.2.2 Requirements Specification

###### 1.2.2.1 Documentation

###### 1.2.2.1.1 Draft information domain

###### 1.2.2.1.2 Draft functions of system

###### 1.2.2.1.3 Draft implementation constraints

###### 1.2.2.1.4 Meet to finalize documentation

#### 1.3 Design

##### 1.3.1 Design Documentation

###### 1.3.2.1 Generate Class Diagram

###### 1.3.2.2 Draft major scenarios

###### 1.3.2.2.1 Draft dynamic model for each scenario

###### 1.3.2.3 Meet to finalize documentation

#### 1.4 Coding

##### 1.4.1 Minimal Increment

##### 1.4.2 Target System

#### 1.5 Testing

##### 1.5.1 Creating unit tests

##### 1.5.2 Test execution

##### 1.5.3 Test Analysis

#### 1.6 Showing

##### 1.6.1 Project Demo

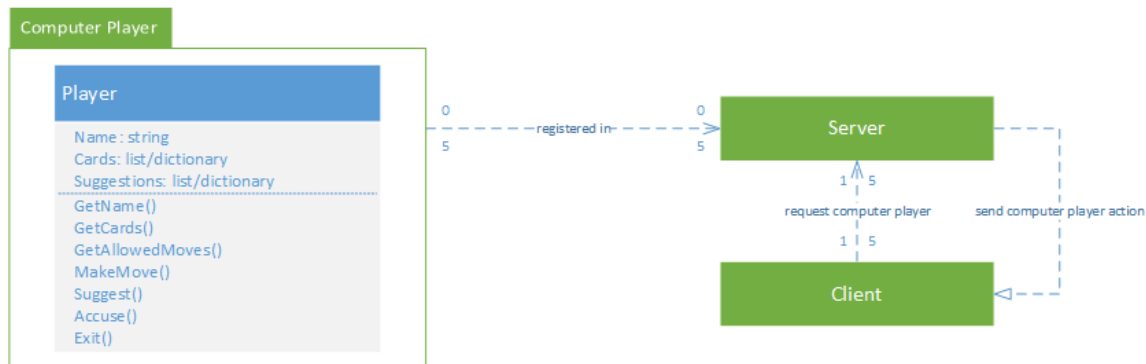
##### 1.6.2 Submit Final versions of Document deliverables

## Vision (Ethan)

Coding a computer client/user for the game of Clue-Less so that a human player is able to add one or more computer user(s) at the beginning of game play. Each computer user will be autonomous and capable of participating in the same way a human player participates.

## Skeletal Architecture

The following UML Class diagram shows the three packages (sub-systems) in the Clue-Less application, Computer Player, Server and Client. Only the Computer Player package is expanded to show the member Player within it. Multiplicity is shown between the Computer Player and Server – 0 to 5 computer players are allowed in the target system. In addition, the client can request between 1 and 5 computer players from the server.



## Minimal Feature Set

The computer player code must provide the following set of minimal features to the Clue-Less client application.

- The client application can add a single computer player to a game.
- The computer player must respond to a “your turn” instruction from the client.
- The response from a “your turn” instruction is to move the computer player following the same rules defined by the client application for human users.
- The computer player will maintain a list of known suggestions. Suggestions might be provided by other players or made by this computer player.
- On a computer player’s turn, the computer player must be able to move into a room, out of a room or between rooms and make suggestions
- When the client application announces the end of the game for any reason (correct accusation or otherwise), the computer player must shutdown gracefully.

## Target Feature Set

In one semester, the minimal set of capabilities outlined previously will be implemented and all supporting documentation outlined for this course will be produced. In addition, we are targeting these additional features:

- When the client application specifies a number of computer players (from 1 to 6), the client application code must start players on the server.
- The computer player makes an accusation when enough information is known.

## Dream/Future Feature Set

Given more time than one semester, the following capabilities could be added to the computer player:

- take-over for a human player who leaves the game

- leverage statistical modeling techniques to improve suggestions and the probability of making an accurate accusation more quickly than a human player
- allow for selecting different levels of difficulty (more skilled or less-skilled computer player)

## Project Schedule

The following project schedule outlines team Automaton's delivery plan. This aligns closely to the WBS.

Task Name	Duration	Start	Finish
<b>1.) Initiation</b>	<b>1 day</b>	<b>Mon 9/22/14</b>	<b>Mon 9/22/14</b>
a.) Project Discussion	1 day	Mon 9/22/14	Mon 9/22/14
b.) Team Allocation	1 day	Mon 9/22/14	Mon 9/22/14
<b>2.) Planning</b>	<b>42 days</b>	<b>Sat 9/6/14</b>	<b>Mon 11/3/14</b>
a.) Creating Team Charter	2 days	Sat 9/6/14	Sun 9/7/14
b.) Project Plan	2 days	Sat 9/6/14	Sun 9/7/14
c.) Requirements Gathering	<b>10 days</b>	<b>Tue 9/23/14</b>	<b>Mon 10/6/14</b>
(i.) Interface Specification	<b>5 days</b>	<b>Tue 9/23/14</b>	<b>Sun 9/28/14</b>
Documentation	2 days	Fri 9/26/14	Sun 9/28/14
Communication with other Teams	<b>1 day</b>	<b>Tue 9/23/14</b>	<b>Tue 9/23/14</b>
Agreeing on Interface Protocol	1 day	Tue 9/23/14	Tue 9/23/14
(ii.) Requirements Specification	<b>4 days</b>	<b>Wed 10/1/14</b>	<b>Mon 10/6/14</b>
Documentation	<b>2 days</b>	<b>Fri 10/3/14</b>	<b>Mon 10/6/14</b>
Product Backlog Items	2 days	Fri 10/3/14	Mon 10/6/14
Sprint Backlog Items	2 days	Fri 10/3/14	Mon 10/6/14
Functions	<b>1 day</b>	<b>Wed 10/1/14</b>	<b>Wed 10/1/14</b>
Multiplayer	1 day	Wed 10/1/14	Wed 10/1/14
Communication	1 day	Wed 10/1/14	Wed 10/1/14
Suggestions	1 day	Wed 10/1/14	Wed 10/1/14
Accusation	1 day	Wed 10/1/14	Wed 10/1/14
Navigation	1 day	Wed 10/1/14	Wed 10/1/14
(iii.) Constraints	2 days	Thu 10/2/14	Fri 10/3/14
(iv.) Non-functional Requirements	2 days	Thu 10/2/14	Fri 10/3/14
d.) Design	2 days	Fri 10/31/14	Mon 11/3/14
(i.) Associations	2 days	Fri 10/31/14	Mon 11/3/14
(ii.) Aggregation	2 days	Fri 10/31/14	Mon 11/3/14
(iii.) Specialization	2 days	Fri 10/31/14	Mon 11/3/14
<b>3.) Execution</b>	<b>4 days</b>	<b>Wed 11/5/14</b>	<b>Mon 11/10/14</b>
<b>Coding</b>	<b>4 days</b>	<b>Wed 11/5/14</b>	<b>Mon 11/10/14</b>
Minimal Increment	4 days	Wed 11/5/14	Mon 11/10/14
Target System	4 days	Wed 11/5/14	Mon 11/10/14
<b>4.) Monitor and Control</b>	<b>19 days</b>	<b>Wed 11/12/14</b>	<b>Sun 12/7/14</b>
a.) Testing	19 days	Wed 11/12/14	Mon 12/8/14

b.) Debugging	19 days	Wed 11/12/14	Mon 12/8/14
<b>5.) Closure</b>	<b>1 day</b>	<b>Tue 12/9/14</b>	<b>Tue 12/9/14</b>
Project Demo	1 day	Tue 12/9/14	Tue 12/9/14

## Risk Management Plan

This plan is centered on ameliorating possible issues. The two most likely risks are around lack of training and availability of team members. Other events that are unavoidable, such as inclement weather, are listed as generic. The following is a risk table showcasing the probability of each event occurring, the impact of that event on the project and a plan to mitigate each risk as much as possible

Risks	Category	Probability	Impact	Risk Mitigation Action Plan
Lack of training in Python	ST	30%	2	Ethan is the most seasoned programmer of the group and doesn't have the learning curve of us all. Shambahvi has some experience with Python and possibly help in problems with syntax or algorithms. Henoke must also get himself acclimated with Python and testing software long prior to doing testing.
Lack of training in "Sockets"	ST	10%	4	Reading up on communication through sockets should be sufficient for less experienced members like Henoke and Shambahvi.
Only one member with computer science background	ST	1%	2	We have the smallest team in the class. Shambhavi and Henoke will have to take it upon themselves to try to be as involved in coding and design so we can help Ethan the best we can.
Changing Requirements	PD	20%	4	Software requirements are maturing. However, minor additions could be made and won't require much modification of existing project.
Inclement weather leading to loss of power	GE	10%	4	We all will take advantage of our nearest public library to work on any deliverables in the case of loss of power due to any inclement weather.
Inclement weather delaying meeting	GE	5%	4	Face to face meetings can be rescheduled if need be. We also can meet Soccoco and voice chat on skype.

Project Equipment Breakdown(Laptops, Hard drives)	DE	5%	3	Since Ethan is the main coder, the fact he has a new machine takes a lot of risk out of a hardware failure derailing our project. Documents will be saved on Google Drive which will keep documents safe in case of hard drive failure.
Loss of team member	ST	1%	1	Barring unforeseen circumstances, everyone plans to be in the course until the end. In the case a team member is not able to make a meeting, all members agreed on a notification of at least 2 days prior to a scheduled meeting. Emergencies are impossible to plan for however we have scheduled our meetings at least 2 days before the deliverable deadline (meetings are usually Sundays). Henoke and Shambhavi have also been learning the Python language to mitigate potential loss of Ethan not being able to program for some time.
Inter team communication issues	ST	3%	3	In the event that communication between team leaders is a challenge, Ethan will communicate via email with other team members of that group.

**Keys**

GE= Generic  
 DE = Development environment  
 ST = Staff Size and experience  
 PD = Process Design

**Impact**

- 1 Catastrophic
- 2 critical
- 3 marginal
- 4 negligible

**Quality Plan – Purpose and Scope**

The plan specifies the practices, standards and checkpoints which will be implemented to build the computer player for the game Clue-less.

## Quality Assurance

This section outlines quality roles and responsibilities and quality review deliverables.

### Quality Roles and Responsibilities

The team will be responsible for assuring the quality of the deliverables. Specific responsibilities are-

Role	Responsibilities
Project Manager Shambhavi Shankrit	Work on the schedule and collaboration, allocate responsibilities, adjust requirements and follow up on the deliverables and deadlines.
Architect Ethan Wilansky	Decide the technical standards which include the usage of Git/GitHub as the control system, PyCharm CE for development work, Python3 as the programming language and Scrum as the Agile software development framework.
SQA Engineer Henoke Shiferaw	Test software program to evaluate the performance, identify issues and bugs, and review the requirements for the computer player.
Configuration Manager Shambhavi Shankrit	Track implementation of the requirements and components for the computer player, status of the development process and the changes included. Trace back the defects.
Developer Ethan Wilansky	Detail the game structure and game play for the computer player. Write the code for the computer player to play the game.
Tester Henoke Shiferaw	Ensure that the computer player is able to play the game and interact with the human player and server.
Technical Writer Shambhavi Shankrit	Document the technical details specifying plans, requirements and deliverables after discussion with the team members.

### Quality Review Deliverables

- Scrum will be used as the Agile software development technique. Requirements will be converted to Product Backlog Items and Sprint Backlog Items.
- Communication between the teams should ensure that interface requirements and specifications amongst each subsystem is clearly defined.
- Documents need to be reviewed to track the progress as per the project plan, interface, requirements and other deliverables.

### Testing approach

The software will be tested using the following approaches:

- **Unit testing-** The framework which will be used for this is Python unittest. This includes features such as test automation, sharing setup and shutdown code for tests, collection of tests and no dependence of the tests on the reporting framework.
- **Integration testing-** This will be performed using local socket server with mock responses to evaluate the dependencies and compatibilities between the computer player, server.



- **Interface testing-** Exchange of messages such as accusations and suggestions by the players and speed of client response will be monitored. The information will be stored in memory for the computer player only.
- **System testing-** The performance of the completely integrated system will be tested to meet the acceptance criteria.

### Configuration Management

This will look into the implementation and track the requirement specific changes and update the document for the deliverables.

- Source control system to maintain and collaborate on the coding effort.
- Packaging applications to deploy to the server and to provide an easy approach for updates.

---

<sup>i</sup> Rossum, Guido van, et al, \*The Python Language Reference\*, Python Software Foundation;  
<https://docs.python.org/3/library/unittest.html>