# 

Innovation Drive and Technology Expo

Project Management Plan

Chris Ciolek

Nicholas Spencer

Maxx Achtman

Apolonio Cazares

# 

[**Introduction**](#_fia60bxdbftx) **4**

[Problem Statement](#_yqj30twfg26x) 4

[Project Scope](#_xvro3vbdan6v) 4

[Inclusions:](#_id7zeeefwhbw) 4

[Inputs:](#_e2q6dobdb20w) 4

[User Input for Concept Submission](#_wsj47k2s1irk) 4

[User Input for Event Registration](#_jvuiz0pdnav2) 4

[Processing Functionality](#_vde0un3ithl3) 5

[Outputs](#_h4ham1co4obr) 5

[Exclusions](#_g7ncnq89qwj) 5

[Major Software Functions](#_1sik65kcf0m4) 5

[Hardware Resources](#_3idejmm09r2y) 6

[Software Resources](#_i4ncg2esa2d5) 6

[Performance / Behavioral Issues](#_7jca8oj9zo2n) 6

[Management and Technical Constraints](#_b46a27vptkc2) 6

[Management Constraints](#_c8l629lklp9u) 6

[Technical Constraints](#_hkmi3u1n1x5l) 7

[Team Organizational Structure](#_k3un0pvs10zw) 7

[**Project Estimates**](#_kcv1co3cv08p) **7**

[Historical Data Used for estimates](#_clqzit3e4fns) 7

[Estimation Techniques Applied and Results](#_mpotpcpfo4r7) 7

[PMP Completion Estimation](#_k6fppojryhio) 7

[Overall Project Estimation](#_83uhux8n0qzl) 8

[Line-of Code Estimate](#_nlksyloo04ih) 8

[Function Estimate](#_9j5jpjgpvyy0) 8

[Total overall project time estimate in hours of effort](#_ke9gt27ixw8k) 8

[Estimation techniques applied and results](#_uit6bz9lgn67) 8

[Estimation Technique 1 - Lines of Code](#_9nqlyk4gt2tk) 8

[Estimation for Technique 1 - Lines of Code](#_639mv1eo2e9w) 8

[Estimation Technique 2 - Function Points](#_54pcr4o635t8) 9

[Estimate of Technique 2 - Function Points](#_kus95cjesa3e) 9

[Estimation technique 3 - COCOMO](#_e61vaoceiy5v) 9

[Estimate for Technique 3 - Final](#_epniw34lxobv) 10

[Project Resources](#_mpotpcpfo4r7) 10

[Budget Estimation](#_mpotpcpfo4r7) 11

[**Risk Management**](#_2m3dtmrg7a26) **11**

[Project Risks](#_dy4i9k4929t) 11

[Risk Table](#_5p4kdmfnf1h4) 11

[Overview of Risk Mitigation, Monitoring, Management](#_v32vtolge7r3) 13

[**Project Schedule**](#_5ejbmlnzcxw4) **13**

[Project Task Set](#_ko35crlpxyy2) 13

[Functional Decomposition](#_ko35crlpxyy2) 15

[Task Network Diagram](#_ko35crlpxyy2) 17

[Timeline Chart](#_ko35crlpxyy2) 18

[Project Schedule](#_ko35crlpxyy2) 18

[**Staff Organization**](#_xu7172j1fj5c) **18**

[Team Structure](#_1k5l25sshj4x) 18

[Management Reporting and Communication](#_1sfiu3e0422d) 19

[**Tracking and Control Mechanisms**](#_go4kylh4v8wn) **19**

[Quality Assurance and Control](#_5tv0n1337v87) 19

[Change Management and Control](#_ueifuuhbejyd) 19

[Tools Used](#_8pr71eh7l3lr) 19

[**Appendices**](#_u9pwhb71hpxd) **20**

[Product Models](#_8ear2tozzzc7) 20

[Process Models](#_8ear2tozzzc7) 21

# 

# 

# 

# 

# 

# Introduction

## Problem Statement

Design an application that can be used for the Innovative Drive and Technology Expo held by Ford every year. The application would allow a user to submit a concept and register as an attendee for days of the event. On the days an attendee would show up, they would be able to scan a QR code and receive a badge.

## Project Scope

### Inclusions:

#### Inputs:

##### User Input for Concept Submission

* Concept Title
* Concept Description
* Strategic Wheel
  + Technology Category
* Concept Type
  + Drive-able
  + Static vehicle
  + Buck
  + Part(s)
* Shipping From
  + City
  + Country
* Source
  + Ford
  + Supplier
* Ford Contact
* Ford Presenter
* Director
* Supplier Company
* Comments

##### User Input for Event Registration

* UserID
* Email
* First Name
* Last Name
* Nickname
* Phone Number
* Cell Number
* Company
* City
* Country
* Technology
* Dates to attend event

#### Processing Functionality

* Database for Concept Submission
* Database for Event Registration
* Send attendee email confirmation
* Generate Excel document containing Concept Submission for Ford Management
* Digital waiver is signed

#### Outputs

* QR Code for attendee
* Confirmation email
* Password for attendee to edit information
* Printable badge on event days
* List of people attending a specific day of event?
* Badge sheet

### Exclusions

* N/A

## Major Software Functions

* Add a Concept to the Concept Submission database
* Add an attendee to the Event Registration database
* Generate an excel file that would be used for Ford Management that holds all concepts submitted
* Allow editing of Event Registration for individual user
* Admin user can add Concepts after submission deadline
* Admin user can add attendee after registration deadline
* A digital waiver would need to be signed before an attendee can register
* Fill out Concept Submission page
* Fill out Event Registration page
* Print Attendee badges
* Send email confirmation
* Open and close Event Registration page
* Open and close Concept Submission page
* Update technology category
* Update technology
* Generate badge sheet
* Generate QR code

### Hardware Resources

* Laptop
* Camera
* Printer

### Software Resources

* Ford Internet
* Java Compiler
* SqlLite Compiler
* HTML, CSS, and JavaScript Compiler
* Database for Concept Technology
* Database for Event Registration

## Performance / Behavioral Issues

* Work in confines of Ford network

## Management and Technical Constraints

### Management Constraints

* Work
* Transportation
* Weather
* Thanksgiving Break
* Vacation time
* Ethics Project
* Due Dates:
  + Project Management Plan: November 6th
  + Software Quality Assurance: November 20th
  + Project Prototype: December 4th
* General Availability:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Sun** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Sat** |
| 8:00 AM |  |  |  |  |  |  |  |
| 9:00 AM |  |  |  |  |  |  |  |
| 10:00 AM |  |  |  |  |  |  |  |
| 11:00 AM |  |  |  |  |  |  |  |
| 12:00 PM |  |  |  |  |  |  |  |
| 1:00 PM |  |  |  |  |  |  |  |
| 2:00 PM |  |  |  |  |  |  |  |
| 3:00 PM |  |  |  |  |  |  |  |
| 4:00 PM |  |  |  |  |  |  |  |
| 5:00 PM |  |  |  |  |  |  |  |
| 6:00 PM |  |  |  |  |  |  |  |
| 7:00 PM |  |  |  |  |  |  |  |
| 8:00 PM |  |  |  |  |  |  |  |
| 9:00 PM |  |  |  |  |  |  |  |

### Technical Constraints

* GitHub
* Size of database
* Protection underneath Ford Internet
* Limited Programming Languages

## Team Organizational Structure

* Lateral Organizational Structure
  + Focus on Teamwork
  + Split tasks according to people’s strengths

# Project Estimates

## Historical Data Used for estimates

* School Projects
* Work Projects
* Vacation Time
* Weather Conditions
* School Scheduling
* Work Scheduling
* Meetings with client
* SLOC/FP Data from qsm.com
* FP calculator from University of Southern California

## Estimation Techniques Applied and Results

### PMP Completion Estimation

* 100 Hours

### Overall Project Estimation

#### Line-of Code Estimate

* 6527 SLOC average

#### Function Estimate

* 184 Function Points

#### Total overall project time estimate in hours of effort

* Effort: 24.524 person-months
* ~3923 hours
* Per person:
* 6.131 person months, 980.75 hours

## Estimation techniques applied and results

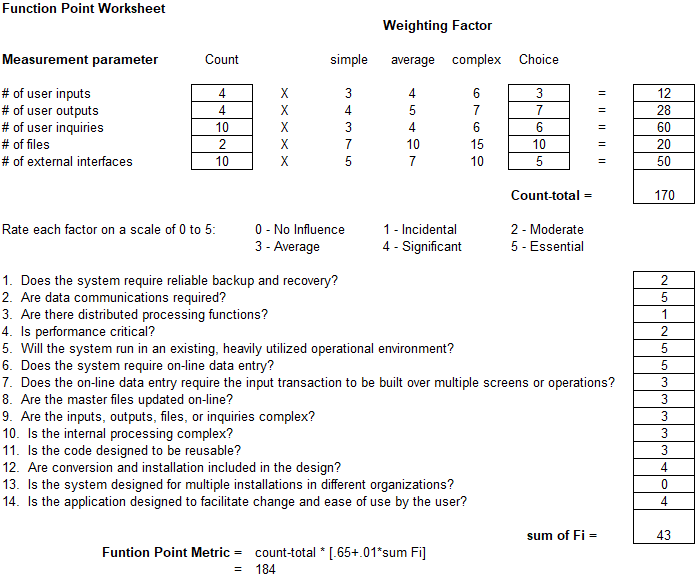
### Estimation Technique 1 - Lines of Code

* Several components to track, including:
* Front end React code, ~100 LOC/page, ~50 LOC/React Component. Estimated ~25 webpages, ~10 React components.  
    
  Front end estimation = 100\*25 + 50\*10 = 3000 LOC
* Server end Nodejs Code, ~30 LOC/CRUD operation, ~150 LOC/database object, ~200 LOC session management, ~200 LOC validation, ~12 CRUD ops, ~3 database objects  
    
  Server side estimation = 30 \* 12 + 150 \* 3 + 200 + 200 = 1210 LOC
* Configuration files for node, babel, and webpack, ~100 LOC NPM config, ~150 LOC webpack config, ~20 LOC babel config  
    
  Configuration estimation = 100 + 150 + 20 = 270 LOC
* Database SQL scripts, ~70 LOC user table, ~40 LOC technology table  
    
  Database estimation = 70 + 40 = 110 LOC

### Estimation for Technique 1 - Lines of Code

* Total estimation is as follows:  
    
  Front end + server end + configuration + database scripts  
  3000 LOC + 1210 LOC + 270 LOC + 110 LOC  
    
  Final Estimation = 4590 LOC

### Estimation Technique 2 - Function Points



* Total amount of function points calculated using calculator created by University of Southern California

### Estimate of Technique 2 - Function Points

* Total number of SLOC using function points and JavaScript as a reference according to QSM resources, is as follows:  
    
  184 \* 46 = 8464

### Estimation technique 3 - COCOMO

* To recap, a component evaluation of SLOC resulted in an estimated 4590 SLOC
* A function point evaluation for SLOC resulted in an estimated 8464 SLOC
* Don’t reconcile well, average is 6527 SLOC
* COCOMO equation is as follows:
* E is the resulting effort required in person-months, a is 3.0 for semi detached, and b is 1.12. KLOC is the line of code total in thousands.
* COCOMO result for LOC evaluation -> E = 16.536
* Result for FP evaluation -> E = 32.809
* Average evaluation -> E = 24.524
* We will continue with the average evaluation

### Estimate for Technique 3 - Final

* Average COCOMO evaluation for effort required was 24.524 person-months
* With a 4 man team, that is 6.131 Months Required for scheduling
* Wage per person at $75/hour over 6.131 months results in $73,572 per person
* 4 people on team, total cost from wages is $294,288

## Project Resources

* + 1. People
       - Contacts at Ford to help with verification and deployment to Ford servers
    2. Hardware
       - (Potentially) SuSE server at Ford for development deployment and final production deployment
    3. Software
       - VSCode for text editing
       - Nodejs for server side scripting
       - Reactjs/Angularjs for client side rendering and routing
       - NPM for package management and development scripts
       - Babel7 for ES6 JavaScript compilation
       - Webpack for creating deployment ready assets
       - Windows/MAC operating systems for developing on
       - SuSE operating system for deployment
       - WebSphere WAS for deployment
    4. Tools
       - Personal computers and laptops for development

## Budget Estimation

* + 1. As pulled from 2.3.6, it will cost the project $294,288 for completion of project in allotted time
       - **Note:** we are excluding production costs such as databases and servers

# Risk Management

## Project Risks

* Poor Estimation on functionalities added during project creation
* Poor Estimation on Main Functionality
* Project Specifications changes
* Underestimation of Program Complexity
* Absence of Team Leader
* Absence of Other Team Members
* Inclement Weather
* Appearance of new technologies needed for functionalities
* Absence of Client

## Risk Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Probability | Impact | Mitigation Plan | Contingency Plan |
| Poor Estimation on Functionalities added during project creation | Medium | Low | Get regular updates and design main functionality to not rely on any additional functionalities | Drop support and expectations for respectable additional functionalities |
| Poor Estimation on Main Functionality | Medium | High | Get regular updates from team members to gauge progress, re-evaluate estimations regularly | Drop additional functionalities and focus on main |
| Project Specification changes | High | Medium | Focus on designing the project instead of programming it, design components modularly so if something changes the whole project doesn’t have to be redone | Change components that are affected by specification change |
| Underestimation of program complexity | Medium | Medium | Keep program functionality simple, have algorithms planned before creation of code | Reduce or remove functionality causing problems |
| Absence of Team Leader | Low | Medium | Finish Planning as early as possible, stay in contact | Assign new team leader and work off of what we have |
| Absence of Other Team Members | Low | Medium | Communicate regularly | Pick up the slack from other team members if items start falling behind |
| Inclement weather | Medium | Low | Adhere to schedule | Attempt digital meeting |
| Appearance of new technologies needed for functionalities | Medium | Medium | Attempt to learn early | Redesign plan to omit new technologies |
| Absence of Client | Low | Medium | Communicate regularly with client and raise questions immediately | Continue to focus on tasks that we know of for the project and wait until more news is brought up form the client |

## Overview of Risk Mitigation, Monitoring, Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Impact | | | | |
| Probability |  | Low | Medium | High | Critical |
| Low |  |  |  |  |
| Medium |  |  |  |  |
| High |  |  |  |  |
| Legend | | | | | |
| Monitor | | Execute Mitigation Plan | | Execute Contingency Plan | |

* Re-evaluate risks once every one-two weeks

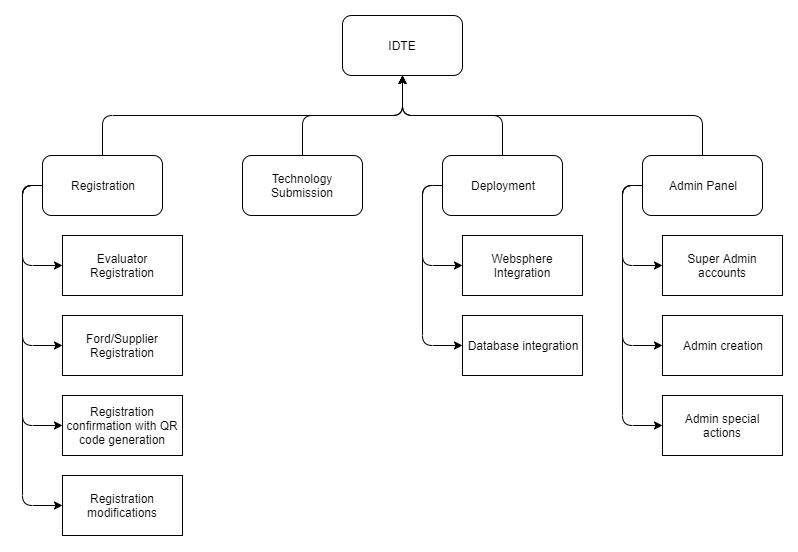
# Project Schedule

## Project Task Set

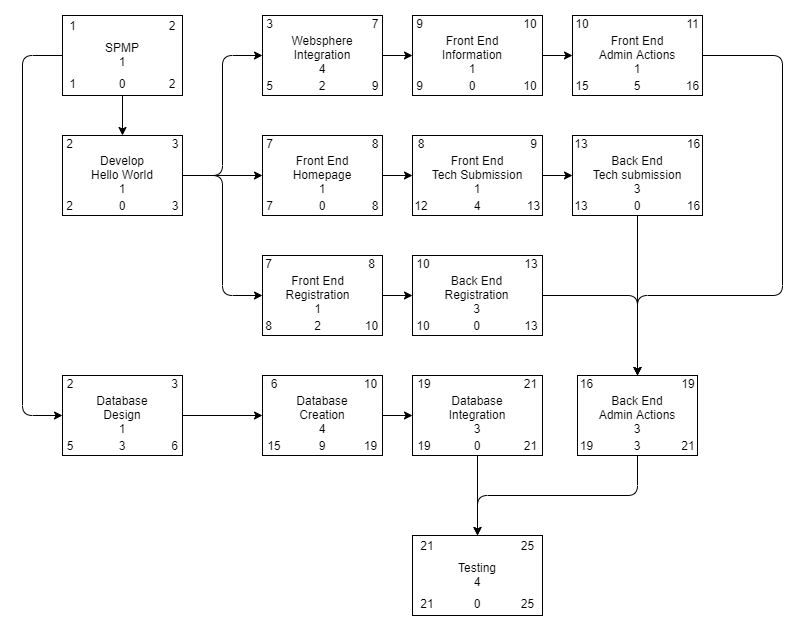
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Task | Time  (Weeks) | Predecessor Tasks | Successor Tasks | Start Date (ES) | End Date (EF) | Slack Time | % Complete |
| SPMP | 1 |  | Develop Hello world | 1 | 2 | 0 | %100 |
| Develop Hello World | 1 | SPMP | Websphere Node Integration | 2 | 3 | 0 | %100 |
| Websphere Node.js integration | 4 | Hello world | Testing | 3 | 7 | 2 | %100 |
| Front End Home page | 1 | Hello world | Front End Tech Submission | 7 | 8 | 0 | %100 |
| Front End - Registration | 1 | Hello world | Back End - Registration | 7 | 8 | 2 | %100 |
| Front End - Technology submission | 1 | Hello world | Back End - Technology submission | 8 | 9 | 4 | %100 |
| Front End - Information | 1 | Websphere  Integration | Testing | 9 | 10 | 0 | %100 |
| Front End - Admin Actions | 1 | Hello world | Front End - Admin Actions | 10 | 11 | 5 | %100 |
| Back End - Registration | 3 | Front End - Registration | Testing | 10 | 13 | 0 | %100 |
| Back End - Technology submission | 3 | Front End - Technology submission | Testing | 13 | 16 | 0 | %100 |
| Back End - Admin Actions | 3 | Front End - Admin Actions | Testing | 16 | 19 | 3 | %100 |
| Database Design | 1 | SPMP | Database Creation | 2 | 3 | 3 | %100 |
| Database Creation | 2 | Database Design | Database Integration | 6 | 10 | 9 | %100 |
| Database  Integration | 3 | Database Creation  Back end registration | Testing | 19 | 21 | 0 | %100 |
| Testing | 4 | All Back end tasks | Deploy | 21 | 25 | 0 | %100 |

## Functional Decomposition

A functional breakdown to be used for scheduling is presented here.



## Task Network Diagram



## Timeline Chart

### Project Schedule

# Staff Organization

## Team Structure

* Maxx Achtman
  + Team Lead
  + Tester
  + Spokesperson
  + Diagram Designer
* Christopher Ciolek
  + Coordination
  + Programming
  + Bookkeeping
* Apolonio Cazares
  + Analysis
  + Programming
  + Backend Design
* Nicholas Spencer
  + Programming
  + UI Design
  + Document Re-visioning

\*changes may be made during project life cycle

## Management Reporting and Communication

* Discord
* Google Drive
* Canvas
* In Person
* GitHub
* Email

# Tracking and Control Mechanisms

## Quality Assurance and Control

* All classes will have test stub evaluations before implementation
* Each group member will carry out testing on both their own and others’ work
* A designated group member will conduct final testing

## Change Management and Control

* Project artifacts will be tracked and maintained using a GitHub repository
* Proposed changes should be submitted with a brief but descriptive summary of what was changed, i.e. added, removed, or modified and explain why
* Changes to artifacts will be announced through the discord server and the group may vote to revert the changes

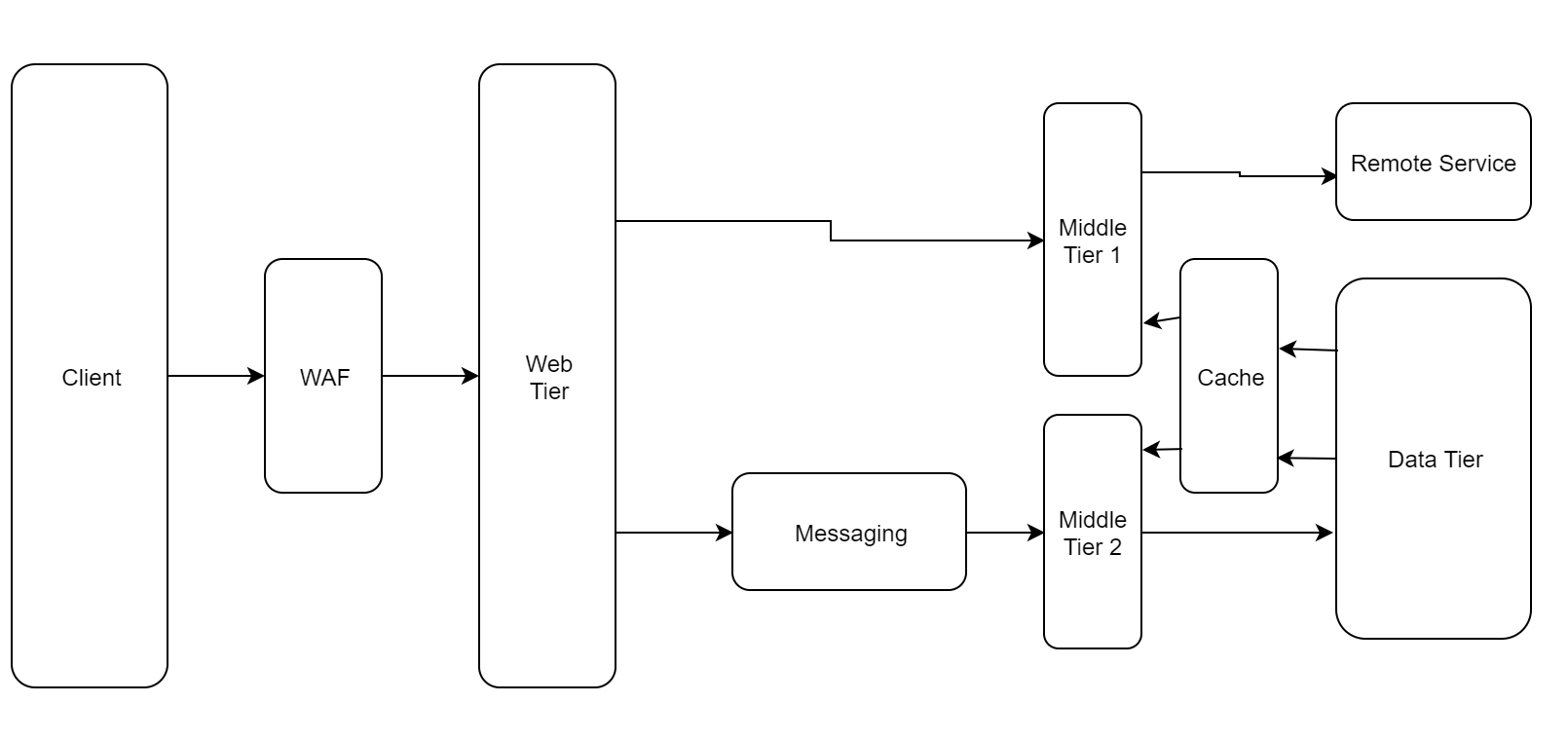
## Tools Used

* GitHub
* Discord
* Visual Studio
* Google Drive
* Enterprise Architect

# Appendices

## Product Models

Due to the nature of being a web application, we chose to use the N-tier architecture for this software.



## Process Models

The process model we used was mainly used was the waterfall model, going to the next phase only after the previous one has been completely finished. Meeting weekly with the client ensures each phase passes with as little friction as possible. Currently we’re in the middle of the design phase clarifying how the software will be built, and getting our development environment set up.

