Software Project Plan

**MoonEyes: Case Management System for Metro Detective Agency**

Senior Design I Summer 2023

Team Members:

Kylie Callison

Erika Valle-Baird

Kendall Gesek

Pamela Shahu

Table of Contents

[**1.0 Introduction 3**](#_heading=)

[1.1 Project scope 3](#_heading=h.z2azgigmifj3)

[1.2 Major software functions 4](#_heading=h.1ikbmswljytl)

[1.3 Performance/Behavior issues 4](#_heading=h.eoj3h59d7j56)

[1.4 Management and technical constraints 5](#_heading=h.wtpahddkfcfb)

[**2.0 Project Estimates 6**](#_heading=)

[2.1 Historical data used for estimates 6](#_heading=h.qq8x2wiadi6e)

[2.2 Estimation techniques applied and results 7](#_heading=h.ucx7bfkmgdxw)

[2.2.1 Estimation technique COCOMO Function Point-Based Estimation 7](#_heading=h.eunlqi3ke1s8)

[2.2.2 Estimate for technique COCOMO Function Point-Based Estimation 8](#_heading=h.busivaey4rt9)

[2.2.1 Estimation technique Process-Based Estimation 9](#_heading=h.siq0oa6tjrmb)

[2.2.2 Estimate for technique Process-Based Estimation 9](#_heading=h.lx7q60jd9k1n)

[2.3 Reconciled Estimate 10](#_heading=h.y6vi6hlpj6c9)

[2.4 Project Resources 10](#_heading=h.kjc6tdr6wiqm)

[**3.0 Risk Management 11**](#_heading=)

[3.1 Project Risks 11](#_heading=h.p3liqlwjbs5j)

[3.2 Risk Table 11](#_heading=h.sni7cfxa211f)

[3.3 Overview of Risk Mitigation, Monitoring, Management 12](#_heading=h.5f728gb2k23k)

[**4.0 Project Schedule 14**](#_heading=)

[4.1 Deliverable and Milestones 14](#_heading=h.p25ddxfero8a)

[4.2 Functional decomposition 15](#_heading=h.mrfegyadby0d)

[4.3 Task network 17](#_heading=h.8cu5aigkh57w)

[4.4 Timeline chart 18](#_heading=h.z3tueif5h8ut)

[**5.0 Staff Organization 19**](#_heading=)

[5.1 Team structure 19](#_heading=h.awwo8kduj9s4)

[5.2 Management reporting and communication 20](#_heading=h.7vf7iwqxwe98)

[**6.0 Tracking and Control Mechanisms 21**](#_heading=)

[6.1 Quality assurance and control 21](#_heading=h.tzk531ekhx6m)

[6.2 Change management and control 22](#_heading=h.opd6772obijc)

[**7.0 Appendix 25**](#_heading=h.3a33li2afas0)

[7.1 Traceability Matrix 25](#_heading=h.yrmbfzg770f9)

[7.2 References 25](#_heading=h.26hxf05pm7aw)

# 1.0 Introduction

## 1.1 Project scope

The software is to serve as a case management system for agents within the Metro Detective Agency. It will allow the agents the ability to monitor cases more efficiently and

The following requirements were given to us for MoonEyes:

* A way in which agents can log in with credentials
* A way in which agents can remotely access the system
* A way in which an agent can enter information about a client, case, subject or agent into a database.
* Ability to search for any parameter within the database
* A way in which data that was previously entered into the database can be clearly viewed
* A way in which agents can view recent submissions or changes
* A way in which an admin user can view audit logs
* A way in which PDF reports can be generated to contain specific case/client/subject information
* A way in which a mass email can be generated and sent out to a subset of clients

**Database Creation**

In order to accommodate the requirements for client, case, agent and subject tracking, a database will need to be created. Specific details that need to be tracked for each table are described below. Furthermore, this is also the information that agents should be allowed to enter in forms and be stored in the database to search. These are additional requirements that specify the type of information that will be submitted within the database and also the parameters for the search functionality.

* **Client Tracking**

Client name, address, phone number, attorney information, and additional notes.

* **Case Tracking**

Case numbers, Purpose, date, reports, photos/videos, and additional notes.

* **Subject Tracking**

Subject name, associates, phone number, place of work, vehicle information (license plate number, type of car), lawyer information, locations visited, repeated locations, background reports, photos/videos, and any other additional notes.

* **Agent Tracking**

Badge number and case numbers.

## 1.2 Major software functions

**Login**

A function to allow users to log into the system using credentials such as a username and password.

**Input**

The ability for users to enter information into the system about cases, clients, and subjects in the proper format.

**Search**

The capacity to search any parameter related to a case, and have the proper case it is attached to returned.

**View**

A user-friendly interface to view information related to the cases, clients, and subjects.

**Email**

For the system to have a mass marketing integration. This will allow the user to send emails to clients from certain search parameters.

**PDF**

A function that will allow the user to download and print certain database information regarding a client, subject, or case.

**Database Management**

The capability to manage all the information entered into the database to ensure the accuracy of the database. Also, the ability to audit changes to the database by the user who made the changes and at what time.

## 1.3 Performance/Behavior issues

**Deployment**

This software will need to be accessible to agents with remote access, so we will need to deploy onto a cloud-based web hosting service such as Amazon EC2. We must make sure all developments are compatible with this service. Additionally, we must ensure that any other services we might use, such as the database platform (MySQL) are compatible with the hosting service as well. Also, it is likely that users will need to be able to use the system simultaneously. Currently, we do not believe many agents will have access to the system at once, since there is only 3 to 4 agents at this time, but expanding the amount of users could cause issues which we need to address during development.

**Security**

This system will be storing personal and sensitive information about clients and subjects. Security is an important consideration that needs to be integrated into the design. This will be addressed in development with secure log in and log out screens. As well as the admin having access to audit logs to oversee any changes being made to the system.

## 1.4 Management and technical constraints

**Funding**

MoonEyes will need to be deployed on a cloud-based hosting service. This will cost the client anywhere from $10 to $50 a month. There could also be additional fees as the data and the company grows.

**Time**

This project has a strict eight-month development timeline. This will impact any improvements or additional features since the timeline is limited to begin with. Since the project is being developed by students, all members of the team will also have to balance other class and work responsibilities, which can greatly affect the time spent on the project compared to a full-time professional software development team.

**Experience**

The team consists of students with limited professional experience in software development. Because of this, the team will rely heavily on independent research, guidance from resources, and lack some of the skills a professional development team would have.

# 2.0 Project Estimates

## 2.1 Historical data used for estimates

Because this project is for a student senior design project, we will not receive any payment. However, these cost estimates will still be calculated in order to fulfill this portion of the document. There also was no existing project to base cost information on.

We based this estimation on labor cost estimates in the U.S. Bureau of Labor Statistics’s “Annual mean wage of software developers, by state, May 2022” data for Michigan, which is displayed below:

| **Definition: Software Developers**  Research, design, and develop computer and network software or specialized utility programs. Analyze user needs and develop software solutions, applying principles and techniques of computer science, engineering, and mathematical analysis. Update software or enhance existing software capabilities. May work with computer hardware engineers to integrate hardware and software systems, and develop specifications and performance requirements. May maintain databases within an application area, working individually or coordinating database development as part of a team  **Michigan**  Mean wage  Annual: $101,800  Hourly: $48.94  Percentile wages   | **10th** | **25th** | **50th** | **75th** | **90th** | | --- | --- | --- | --- | --- | | $61,090 | $80,130 | $101,070 | $126,420 | $140,120 |   Data: U.S. Bureau of Labor Statistics May 2022 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

For this estimation we will use the 25th percentile cost estimation because we are all inexperienced. The labor rate is:

**$((80,130/12)(1+1+1+1) /4) ≈ $6,667**

However, this data does not represent the average *burdened* labor rate. Burdened labor rate can be up to 1.5-2.5x as much as a normal labor rate, so to obtain accurate estimation results we used a burdened labor rate of $8000/month which is reasonable as compared to the U.S. Bureau of Labor Statistics rate.

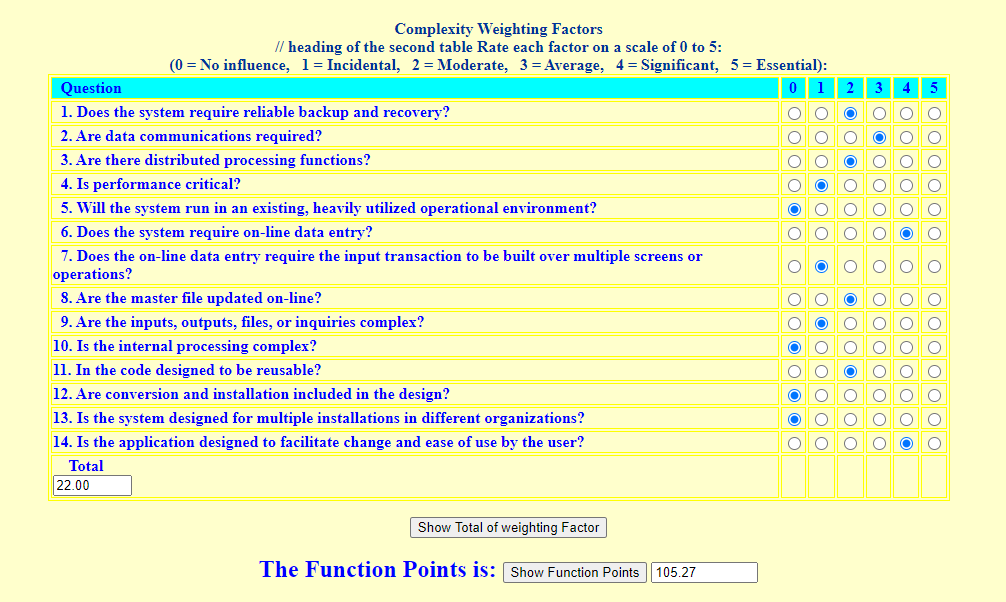
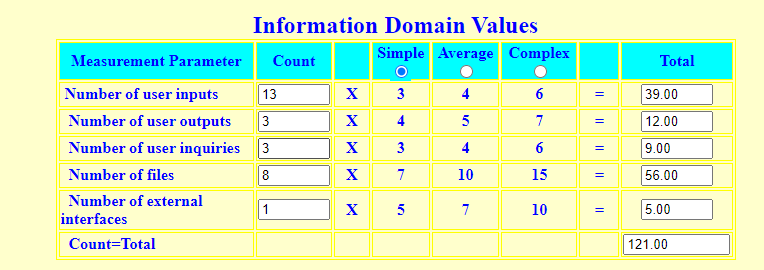
## 

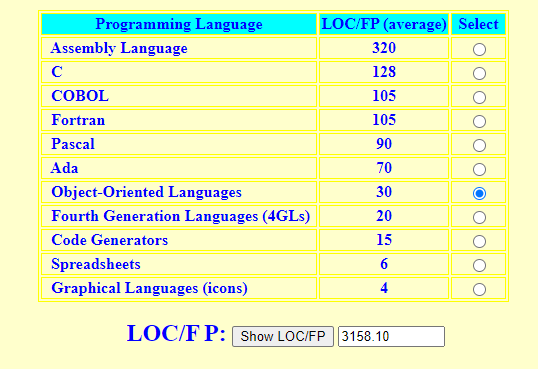
## 2.2 Estimation techniques applied and results

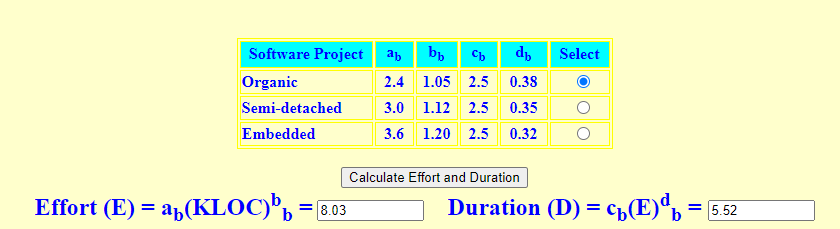
We used two techniques to estimate the person-months and project costs for Moon Eyes.

* **COCOMO Function Point-Based Estimation**
  + This technique was requested. This type of estimation focuses on information domain values over functions.
* **Process-Based Estimation**
  + This estimation technique is based on the processes required in the project. We decomposed MoonEyes into functions and estimated the effort required.

### 2.2.1 Estimation technique COCOMO Function Point-Based Estimation

We used the Tinytool [Function Point Plus Basic COCOMO Calculator (Mohamed & Fouani)](http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/gamel/introduction.html) in order to obtain the estimation for the Function Point technique.





The information domain values were based on the likely actions of a user and not every single input field. Many input fields will be repeated across the database.

*Software Engineering: A Practitioner's Approach* states:

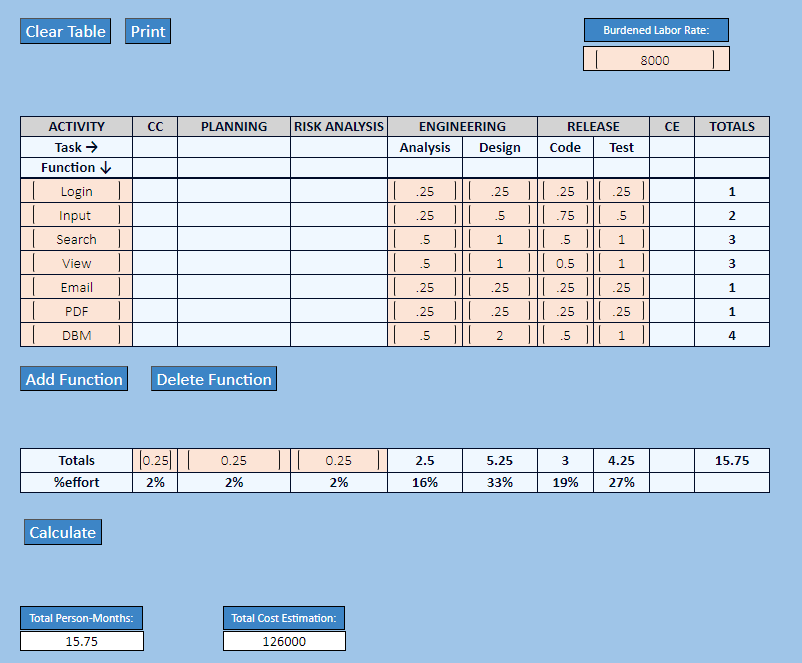
The organizational average productivity for some systems is 6.5 FP/pm. Based on a burdened labor rate of $8,000 per month, the cost per FP is approximately $1,230.

### 2.2.2 Estimate for technique *COCOMO Function Point-Based Estimation*

Based on the calculation of 105.27 Function Points and a cost of $1230/FP. The total cost of the project will be $129,482 and development will take 16 person months. A total of 4 person-months for 4 developers. The total estimated LOC is 3158 because we will be using an OOL.

### 2.2.1 Estimation technique *Process-Based Estimation*

We used the Process-Based Estimation Assistant Tinytool (Callison, Salmassi, Almoamen, Dinh & Len) in order to obtain the estimation for the Process-Based estimation technique.



### 2.2.2 Estimate for technique *Process-Based Estimation*

Using a burdened labor rate of $8000, the Total Person-Months would be 15.75 and total cost estimation would be $126,000. Note that this workload will be split between four developers and is not representative of the actual length of the project, but person-months in effort.

## 2.3 Reconciled Estimate

Both estimated techniques had similar results before attempting reconciliation.

The final cost, effort, time (duration) estimate for the project (at this point in time) is presented here. To calculate the reconciled estimate we will average the two results from the estimation methods in 2.2.2:

**LOC:** 3158

**Number of Developers:** 4

**Person Months:** 15.8 months

**Person Months/Developer or Duration of Development:** 3.96

**Project Cost:** $127,741

We find these estimations to be fairly accurate. Although we have an eight month timeline for development, all developers are also working and taking other classes, so we will be unable to dedicate the same amount of time that a normal development team would, so an estimate of 3.96 person-months is acceptable for the total development timeline.

## 2.4 Project Resources

**People**

* **Four Developers**
  + As designated by the senior design class this project will require the four developers listed on this document.

**Hardware**

* **Development**
  + Need PC with internet connection and specifications to withstand the writing, debugging, testing of Moon Eyes within a version control system.
    - 8GB RAM minimum
* **User Client-side**
  + Need PC with internet connection and access to a browser

**Software**

* **Development**
  + Version Control system such as GitHub
  + User license or access to OOL-supported IDE such as Virtual Studio Code
  + Various VSC Extensions (GitHub Pull Requests)

**Other resources**

* **Development & Deployment**
  + The development team will also need access to various tools in order to develop and deploy the software. Because we will be creating a database users will need access to MySQL
  + Additionally, we will be deploying on an Amazon EC2 web server so we will need an AWS account in order to test

# 3.0 Risk Management

## 3.1 Project Risks

Metro Detective Agency currently has eight categories of risks pertaining to our project and subsequent software.

*These include:*

*Business Impact Risks*: The business impact risk for Metro Detective agency involves the production of a product that is not viable to the business. This may include a product that inadvertently harms the business in some way.

*Customer Risks*: When customers fail to participate in timely communication practices with the development team, the project suffers from customer risks.

*Development Risks*: Should the customer or development team fail to provide all information pertaining to tools necessary to build the product, development risks occur.

*Employee Risks*: As the development team are all currently students at the University of Michigan, Dearborn, the project is at risk due to lack of experience in software development.

*Process Risks*: When quality assurance metrics are not adhered to or developed correctly, process risks may occur leading to low product quality.

*Product Size*: Our project suffers from product size risks when erroneous assumptions in product size estimations are made.

*Technology Risk*: The technological risk associated with our project involves the use of antiquated technology and software. When this occurs, the project timeline is at risk, as developing new compatible software may be a lengthy process. In addition, data security poses another threat to the technological risk of our project, requiring immediate attention.

*Financial Risks*: Financial risks may have an impact on our project if there are budget overruns or incorrect cost estimations. This may be detrimental to the project, resulting in failure.

## 3.2 Risk Table

| Risk ID | Risks | Probability | Impact |
| --- | --- | --- | --- |
| BI | Business Impact Risk | Low | High |
| C | Customer Risks | High | High |
| D | Development Risks | Medium | Low |
| E | Employee Risk | Low | High |
| P | Process Risk | Medium | High |
| PS | Product Size | High | High |
| T | Technology Risk | Medium | Medium |
| F | Financial | Medium | High |

## 3.3 Overview of Risk Mitigation, Monitoring, Management

The RMM Can be viewed as a separate document [here](https://docs.google.com/document/d/12l6bkppMVt4PPXpCQ17xP4Ws_-wVl7pX/edit?usp=sharing&ouid=109378746336515960362&rtpof=true&sd=true).

**Overview**

Description of Possible Risks associated with each category

*Business Impact Risks*

The end product is not accepted by the business, as it no longer fits the overall business strategy for the company and therefore is not wanted.

*Customer Risks*

Ineffective communication causes unsatisfactory customer needs and requirements collection.

The wrong interface is developed due to poor customer cooperation.

Inaccuracies or incomplete information in note documentation can lead to incorrect development.

*Development Risks*

Solutions and code generated by inappropriate versions of tools and components are inefficient.

Integration of tools is unfeasible.

*Employee Risks*

Personnel shortfalls due to inadequate skill set for project requirements.

Development of the wrong software functions.

Poor comments in code.

Staff inexperience and lack of training in tools expected to be used in development.

*Process Risks*

Deviation from software engineering standards results in substandard quality of deliverables.

User interface is too complex and is not easily understood or accepted by users.

Quality metrics that should be consistent are not clearly defined, limiting design and product quality.

*Product Size*

Rate of defect repair is underestimated.

Failures of timely deliverables due to incorrect time estimations.

*Technology Risk*

The database used in the system cannot be accessed.

The system suffers from data logging and storage integration issues.

Outdated technology allows for unauthorized system access, system intrusion, or break-ins.

Integration issues occur with wireless transmission.

Component incompatibility is detected.

*Financial Risks*

Low estimations of cost can lead to irrecoverable project failures.

Shortage of financial or personnel resources with improper budget estimations.

# 4.0 Project Schedule

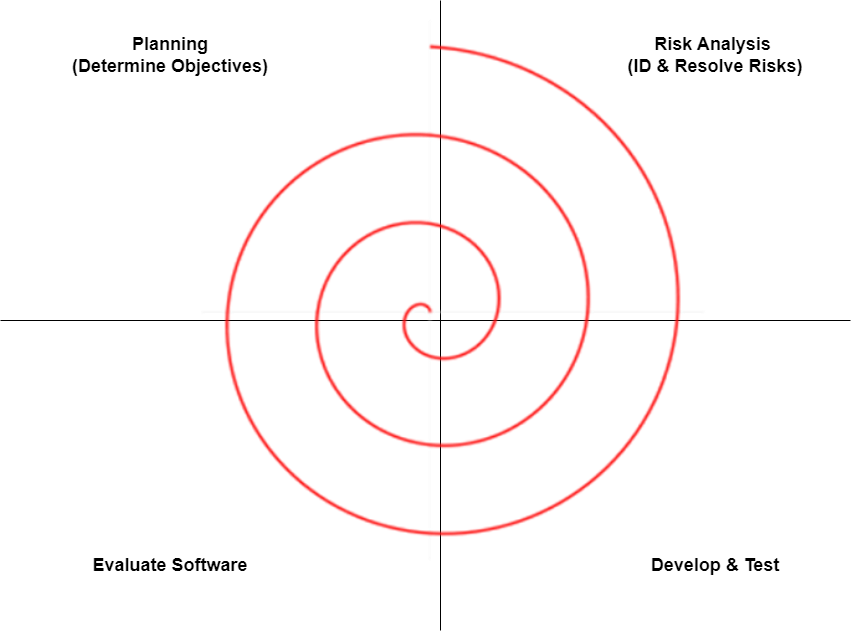
Subsequent information provided contains a directory of scheduled deliverables currently planned for each stage of project MoonEyes development lifecycle.  Included are their corresponding planned completion dates.

## 4.1 Deliverable and Milestones

| **Stage of Development** | **Stage Completion Date** | **Deliverable** | **Deliverable Completion Date** |
| --- | --- | --- | --- |
| Planning | 06/28/2023 | Project Plan  One-Note  ERD Diagram  Data Model  Milestone | 06/28/2023  6/15/2023  6/16/2023  6/17/2023  06/28/2023 |
| Requirements Definition | 07/12/2023 | Use Case Document  Risk Management Plan  Requirements Document  Milestone | 06/07/2023  06/28/2023    07/12/2023  07/12/2023 |
| Design (Functional & System) | 08/09/2023 | Software Quality Assurance Plan  Milestone | 08/09/2023  08/09/2023 |
| Programming | 08/22/2023 | Prototype  Milestone | 08/22/2023  08/22/2023 |

## 4.2 Functional decomposition

Metro Detective Agency software will employ a Spiral Model during design and implementation. This was determined due to client uncertainty, allowing us to develop acceptable solutions as well as allow for stronger emphasis of risk assessment. Weekly check-ins have also been employed to improve communication with our clients. To ensure customer satisfaction, customer requested documentation, unrelated to course work, have been completed for feedback. These additions will help elevate document deadline delays along with allowing us to develop elements for the product as soon as it becomes available. Due to the high uncertainty related via our clients, this model also allows us to accommodate any desired changes.



*Interface Task Breakdown:*

Wireframing

New Project Wizard

Database (DB) Construction

Database Communication with Interface

Exporting PDF File Ability

*Engine Task Breakdown:*

Object Handler

Text Handler

Logic Handler

Attribute Handling

Unit Pathing

File I/O Parser

*Help Task Breakdown:*

Help Task Breakdown

Interface Help

Engine Help

Database Help

FAQ

Manual

*Test Task Breakdown:*

Test Task Breakdown

Unit Testing

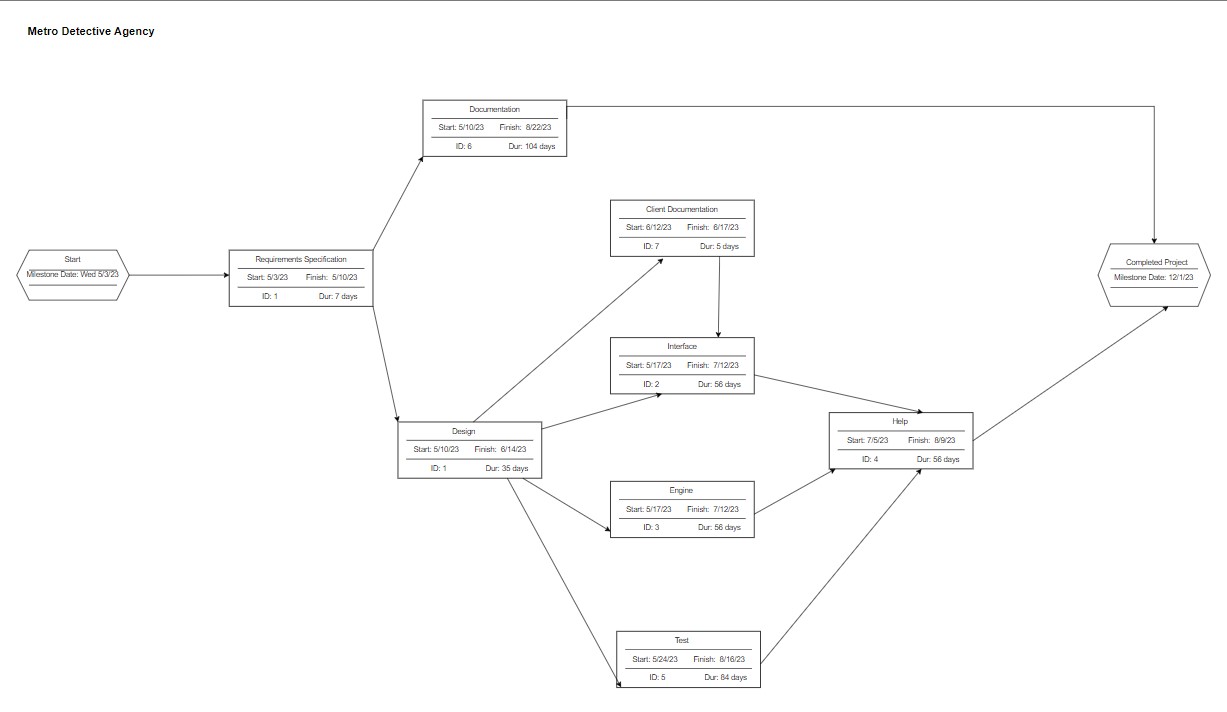
Integration & Database Testing

Validation Testing

Performance Testing

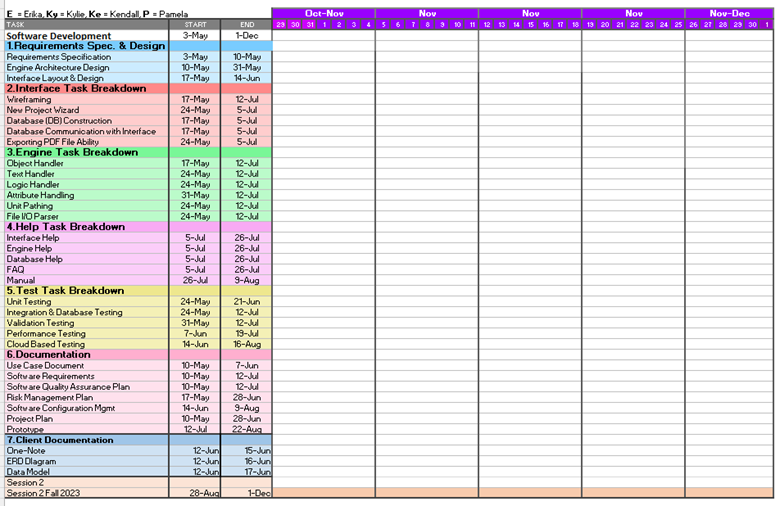
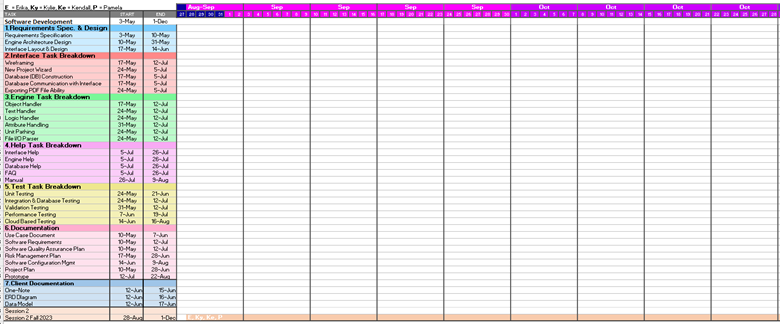
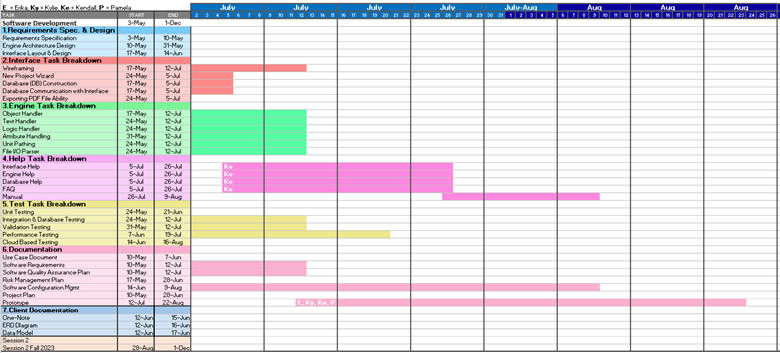
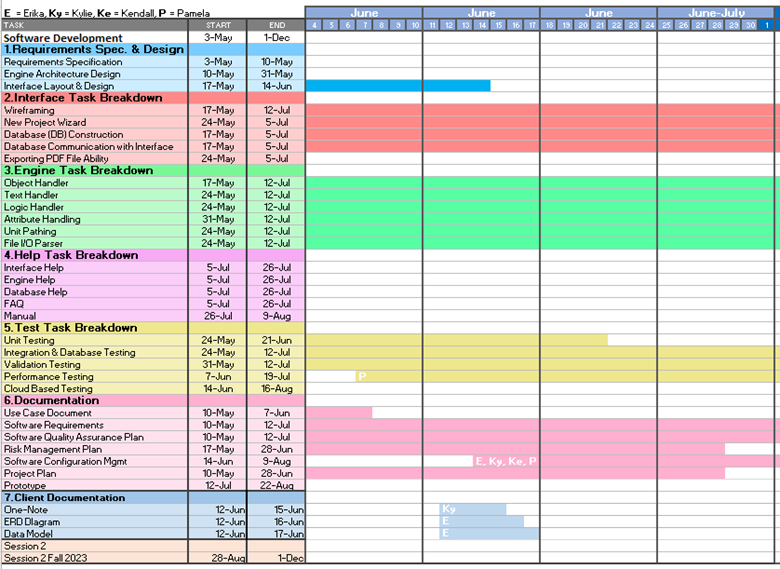
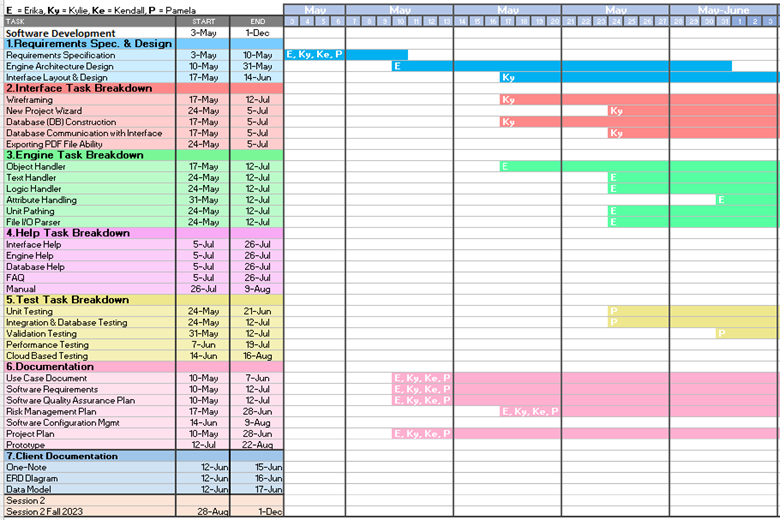
Cloud Based Testing

## 4.3 Task network

Metro Detective Agency’s Task Network, a depiction of project tasks and dependencies they hold, is illustrated below.

## 

## 4.4 Timeline chart

A project timeline chart is presented. This may include a timeline for the entire project or for each staff member. 

# 5.0 Staff Organization

In this section, the structure of the team along with the methods of communicating and reporting are discussed.

## 5.1 Team structure

The team structure for the project will be an egoless and democratic organization. The smaller group size allows for open communication among the team regarding the project decisions, products, and goals.

Database and Advanced Interface Development

· Database Engineering

· Draft Documentation

· Data Model and ERD Diagram

· Engine Architecture Design

· Internal Modules and Advanced Interface Development

· Object, Text, Logic, and Attribute Handling

User Interface Design and Development

· User Interface Development, Layout, & Design

· Draft Documentation

· Prototyping & Wireframing

· User Experience

· Database Communication with Interface

· Cloud Development

Testing, Quality, and Training

· Testing on Unit, Integration, Database, Validation, Performance, and Cloud

· Draft Documentation

· Training Sessions

· Quality Assurance

· Risk Management

· Reports

Documentation Editor

· Proof Reading all Documentation

· User Guides

· FAQ

· Manuals

· All Final Documentation

· Progress Reports

· Assistance in all Development Areas

## 5.2 Management reporting and communication

The team utilizes various types of reporting tools to track the progress on the development of MoonEyes. The clients’ requested the use of a software program called Trello to visualize the progression of the project. Trello is able to divide the project’s user stories into groups from Product Backlog, Planning, In Progress, Review, and Done. It also allows the team to show what is the priority for each of the user stories. All team members and clients have access to the Trello board, and it is reviewed at every client meeting. Within the team, the use of the traceability matrix, Gantt chart, and project schedule with deliverables allows the team to track the progression of the project.

There are weekly meetings between the team and the clients to review the progress of the project. Each meeting has an agenda sent before the meeting begins and a recap sent directly afterward to track the topics of each meeting. Throughout the week, the clients have requested that documents and products be emailed to them to review in order to be more time efficient. The team also communicates throughout the week on the progression of the project and project schedule, as well as the due dates of any deliverables.

# 6.0 Tracking and Control Mechanisms

Techniques to be used for project tracking and control are identified.

## 6.1 Quality assurance and control

*Scope and intent of SQA activities*

The objectives of the SQA team are to ensure project management processes by the development of quality outputs. The SQA team will provide quality assurance by adhering to client design specifications. To guarantee the product meets customer specifications, the SQA team has set up frequent meetings with clients for verification and planning. This was done in an effort to prevent risks that could affect the quality of the final software. Walking through each stage of development with our clients safeguards our software against risks that affect production quality and maintains quality control by sustaining a customer feedback loop. The SQA team sets out to reach its goals of customer satisfaction and defect elimination by monitoring and improving any weaknesses identified in processes. For our team to verify our software is meeting its requirements, adhering to predefined standards, and uncover errors pertaining to function, logic, and implementation, we will be conducting FTRs. FTRs will be conducted via walkthroughs and technical reviews. Due to the size of our team, each team member will employ multi-dimensional roles and will all remain members of the SQA team.

Software Quality Metrics

*Customer Satisfaction*

One of the product quality metrics we plan to employ is customer satisfaction. This metric will be measured using a five-point scale: Very Satisfied, Satisfied, Neutral, Dissatisfied, Very Dissatisfied. To obtain this metric, surveys will be carried out for each aspect of software that is developed.

*Fix Backlog and Backlog Management Index*

This maintenance quality metric will be employed using a Trello board, allowing our team to fix defects based on their priority in the backlog. To increase customer satisfaction, customers have access to the aforementioned board and are able to inform our team about requested fixes at any time, leading our team to maintain shorter fix response times.

*Fix Quality*

Frequent client meetings allow our team to be informed about defects as soon as they are encountered. By eliminating defects as soon as possible, we are able to fix quality, response time, and responsiveness. This quality maintenance metric will be determined by the rate of defective fixes without delinquency.

*Quality Indicators*

To measure and ensure software quality, the SQA team plans to employ the following quality metrics:

Reliability

Portability

Efficiency

Maintainability

Usability

Security

Performance

Further details for the FTR and Quality indicators can be found in the Software Quality Assurance Plan.

## 6.2 Change management and control

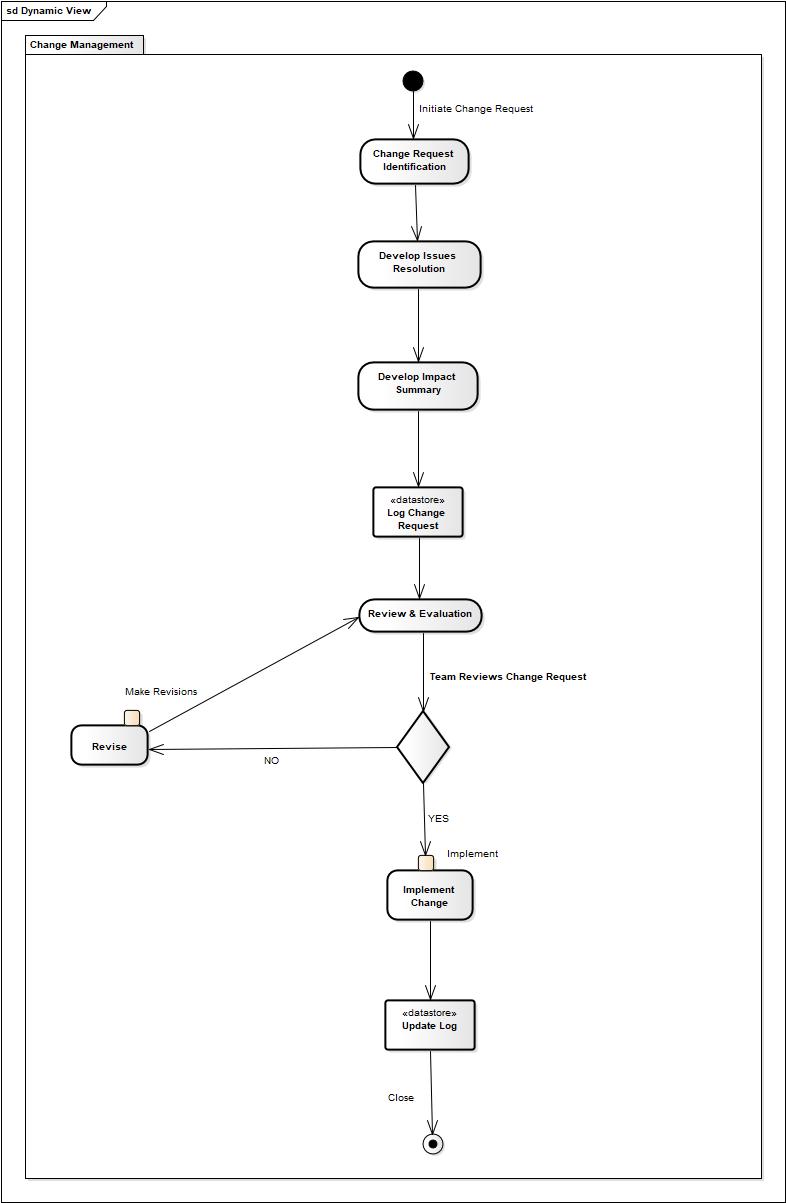
*Scope and intent of SQA activities*

The focal point of the Software Configuration Management (SCM) is the identification and control of major software changes. To ensure proper implementation and uniformity, we will conduct a systematic approach for managing, facilitating, and authorizing changes to our software that verifies our product’s compliance with functional requirements. This approach enables us to provide a product that will meet previously set standards, with the objective of limiting the impact changes can have on the entire system. When a defect is identified in the software pertaining to one of the team’s primary goals, the change request identification process begins, in a methodical fashion, to correct the errors. When this occurs, all changes are evaluated by the entire team and documented for future reference.

*Change Process*

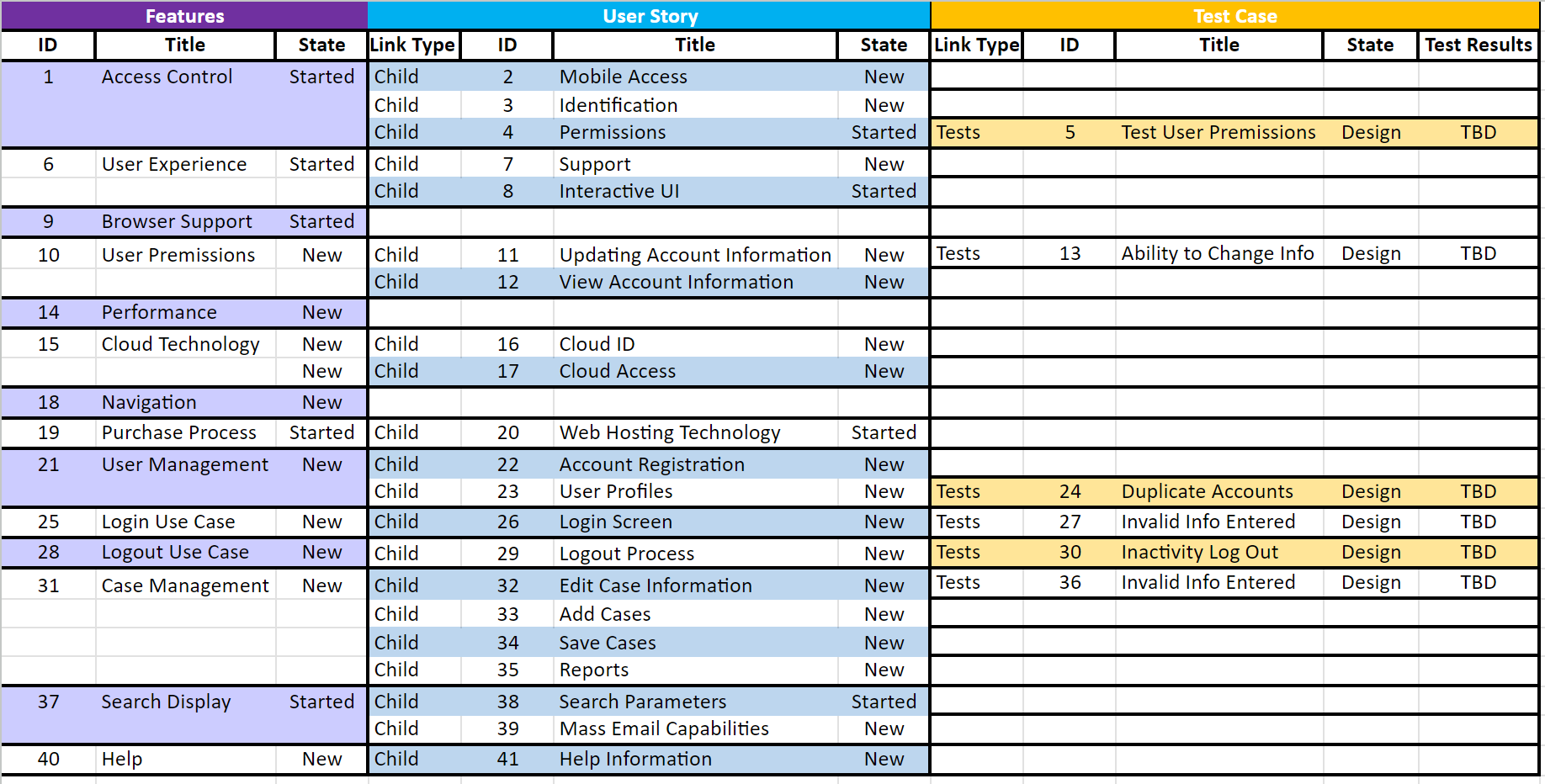
Illustrated below is a graphic depiction of how our team will conduct and employ change management. Beginning with change request identification, both issue resolution and impact summary are required to determine a plan of attack and how this change may affect our system. From this point a change request is logged, to ensure proper documentation. Our team then must review and evaluate such a change, deciding if we will move forward with the change or if the change is not viable. If the change will be beneficial to our program, we then implement the change and update our log. If the change is denied, a revision can be made to then reassess the proposed change as a team at a later time.

Change management plans and processes can be found in the Software Quality Assurance Plan.



# 7.0 Appendix

## 7.1 Traceability Matrix



## 7.2 References

This project plan was created using the latest Pressman format. Additionally, the Cyber Rovers “WMITS Software Design Specification” and “Gameforge Design Document” found on McGraw Hill website were used as reference. The U.S. Bureau of Labor Statistics table depicted with the title “Annual mean wage of software developers, by state, May 2022” was used as reference in section [*2.1 Historical data used for estimates*](https://docs.google.com/document/d/1uNbMmrm_trSYDt_MZm7bvOQ9ErOrXZez08L5keYFHus/edit?pli=1#heading=h.7765v9d1l7uh).

The textbook *Software Engineering: A Practitioner's Approach* was also referenced throughout the creation of this document. Also, multiple tiny tools were used in order to create the cost estimates.

* R. S. Pressman, “Document Templates: Software Project Plan,” *R.S. Pressman & Associates, Inc..* [Online]. Available: <http://www.rspa.com/docs/Projectplan.html>. [Accessed: 14-June-2023].
* “Risk Mitigation, Monitoring, and Management Plan,” *McGraw Hill Higher Education*. [Online]. Available: [www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs1/rmmm.pdf](http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs1/rmmm.pdf). [Accessed: 14-June-2023].
* R. S. Pressman, “Document Templates: RMMM Plan,” *R.S. Pressman & Associates, Inc.*. [Online]. Available: <http://www.rspa.com/docs/riskmmm.html>.[Accessed: 14-June-2023].
* “Software Project Plan,” *McGraw Hill Higher Education*. [Online]. Available: <http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs1/projplan.pdf>.[Accessed: 14-June-2023].
* “Software Project Plan,” *McGraw Hill Higher Education*. [Online]. Available: <https://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs2/projplan.pdf>.[Accessed: 14-June-2023].
* “Occupational Employment and Wages, May 2022: Software Developers”, *U.S. Bureau of Labor Statistics.* [Online]. Available:<https://www.bls.gov/oes/current/oes151252.htm#st>. [Accessed: 14-June-2023].
* Pressman, Roger S., and Bruce R. Maxim. *Software Engineering: A Practitioner's Approach* . Vol. 9, McGraw-Hill Education , 2020.
* Tinytool [Function Point Plus Basic COCOMO Calculator (Mohamed & Fouani)](http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/gamel/introduction.html)
* Tinytool Process-Based Estimation Assistant (Callison, Salmassi, Almoamen, Dinh & Len)